REALSYSTEM PRODUCTION GUIDE

RealSystem Release 8 with RealProducer 8.5

RealNetworks, Inc.
2601 Elliott Avenue, Suite 1000
Seattle, WA 98121
U.S.A.

http://www.real.com
http://www.realnetworks.com

Printed in the United States of America.

CONTENTS

INTRODUCTION

RealSystem™ gives you the power to stream compelling multimedia presentations over a network. It includes RealServer™, the most advanced streaming media server available, along with RealPlayer® and RealPlayer Plus, the world's most popular desktop applications for playing streaming media clips. This production guide will help you produce any multimedia presentation, whether it is a simple video on your home page or a multimedia extravaganza.

Tip

To experience the many possibilities of streaming media, download RealPlayer from **http://www.real.com**, and then visit **http://realguide.real.com**.

How this Book Is Organized

This production guide tells you how to assemble a RealSystem presentation. Although the book provides tips on producing great content, the more you know about producing audio, video, and graphics in general, the faster you will be able to put together a great streaming presentation. If you already know the basics of HTML, you'll find it easy to pick up Synchronized Multimedia Integration Language (SMIL), the language used to assemble RealSystem presentations.

Note

This guide does not explain how to use RealSystem tools such as RealProducer™. For specific information about using a particular tool, refer to the tool's user's guide or online help.

Chapter 1: What's New in RealSystem?

If you're familiar with previous versions of RealSystem, this chapter will give you a quick update on the many changes in this version of RealSystem.

Chapter 2: Planning a Streaming Media Presentation

If you are new to streaming media, this chapter walks you through the steps involved in putting together a RealSystem presentation, explaining bandwidth and timeline concerns.

Chapter 3: Producing Audio

This chapter gives you the background you need to create a RealAudio® file. It then provides pointers on capturing and digitizing high-quality audio content.

Chapter 4: Producing Video

Read this chapter to learn how to capture high-quality video content and optimize it for conversion to streaming RealVideo® clips.

Chapter 5: Producing Animation

Using Macromedia's Flash, you can produce dazzling animated presentations. This chapter explains how to stream Flash clips with RealSystem.

Chapter 6: Assembling a Presentation with SMIL

After you create your multimedia clips, you write a SMIL file that pulls the entire presentation together. This chapter explains how to use SMIL to specify when and how each part of your presentation plays.

Chapter 7: Extending SMIL

RealNetworks® has developed several extensions to SMIL that enhance its media-streaming capabilities. This chapter explains these extensions, which work only with RealServer and RealPlayer.

Chapter 8: Playing Clips in a Web Page

If you want to integrate your presentation seamlessly into your Web page, follow the instructions in this chapter.

Chapter 9: Inserting Ads into a Presentation

This chapter explains how to use SMIL to insert ads into your streaming presentation.

Chapter 10: Delivering a Presentation

This chapter provides step-by-step instructions for moving your streaming clips to RealServer and linking your Web page to them. It also explains how to use a Web server to deliver simple presentations.

Chapter 11: Broadcasting a Presentation

Refer to this chapter if you plan to broadcast an audio or video event.

Appendix A: Quick Answers to Common Questions

If you are new to RealSystem, this appendix answers basic production questions and points you to additional resources on the Internet.

Appendix B: Quick Steps for Streaming Clips

This appendix lists the basic steps you take to get a clip to stream from RealServer or a Web server.

Appendix C: Advanced Production Techniques

For advanced users, this appendix conveys useful production techniques that help you get the most out of RealSystem.

Appendix D: SMIL Quick Reference

Once you understand SMIL as described in Chapter 6, use this appendix as a reference when writing SMIL files.

Appendix E: SMIL Language Codes

If you create streaming clips in different languages, you can use these codes in a SMIL file to indicate language choices.

Appendix F: File Type Reference

This appendix provides a quick reference for common file types used in RealSystem streaming.

Conventions Used in this Book

The following table explains the typographical conventions used in this production guide.

Notational Conventions

Convention	Meaning
variables	Italic text represents variables. Substitute values that are appropriate for your situation.
emphasis	Bold text is used for emphasis.
`[options]`	Square brackets indicate optional values you may or may not need to use.
`choice 1|choice 2`	Vertical lines separate values you can choose between.
…	Ellipses indicate nonessential information omitted from the example.

Additional RealSystem Resources

Most RealNetworks manuals are available in both PDF and HTML formats from the RealNetworks documentation library. The library's main page is at **http://service.real.com/help/library/index.html**. In addition to this production guide, you may need the following resources:

- *RealText Authoring Guide*

  This guide explains how to create streaming text.You can use RealText®, for example, to create a stock ticker or provide video subtitles. Get this book at **http://service.real.com/help/library/encoders.html**.

- *RealPix Authoring Guide*

  With RealPix™ you can create streaming slideshows of still images. *RealPix Authoring Guide* explains how to put a RealPix presentation together with special effects such as fades and zooms.You can find this book at **http://service.real.com/help/library/encoders.html**.

- *RealServer Administration Guide*

  The basic reference for the RealServer administrator, this guide explains how to set up, configure, and run RealServer to stream multimedia. You need this book only if you are running RealServer yourself. It is available at **http://service.real.com/help/library/servers.html**.

- *Embedded RealPlayer Extended Functionality Guide*

  This guide supplements *RealSystem Production Guide*. Available at **http://service.real.com/help/library/encoders.html**, it explains how to use JavaScript or VBScript to control RealPlayer functions for a presentation embedded in a Web page.

- RealSystem Authoring Kit

  The authoring kit contains production tools and manuals in a single, convenient bundle. You can obtain the kit by registering at **http://www.realnetworks.com/products/authkit/index.html**.

- Software Development Kits (SDKs)

  RealNetworks offers SDKs for RealServer, RealPlayer, and RealProducer. Designed for programmers, SDKs help you integrate applications with RealSystem, or create new plug-ins for RealServer and RealPlayer. You can get SDKs by registering at **http://www.realnetworks.com/devzone/downlds/index.html**.

Technical Support

To reach RealNetworks' Technical Support, please fill out the form at:

- **http://customerrelations.real.com/scripts/rnforms/contact_tech_service.asp**

The information you provide in this form will help Technical Support personnel respond promptly. For general information about RealNetworks' Technical Support, visit this Web page:

- **http://service.real.com/help/call.html**

WHAT'S NEW IN REALSYSTEM?

The open, end-to-end architecture of RealSystem gives you more possibilities for creating Web-based multimedia than ever. If you're familiar with previous versions of RealSystem, this chapter gives you a quick look at changes in the latest releases of RealSystem.

What's New in RealSystem 8?

RealSystem 8 builds on RealSystem 7, introducing new features for RealServer 8, RealPlayer 8, and RealProducer 8.5. This section describes new features in RealSystem 8 that affect content authoring.

RealAudio 8 Stereo Codecs

RealProducer 8.5 introduces new RealAudio 8 stereo codecs that create clips with better sound quality, especially at higher bandwidths. Only RealPlayer 8 can play RealAudio 8 clips. For a list of new codecs, see "RealAudio 8 Stereo Music Codecs" on page 37.

New RealVideo 8 Codec

RealProducer 8 introduces the RealVideo 8 codec, which creates RealVideo clips that have significantly better visual quality than clips encoded with previous codecs. Only RealPlayer 8 can play RealVideo 8 clips. For more information, see "RealVideo 8 Codec" on page 53.

Flash 3 and Flash 4 Support

RealServer 8 can stream Flash 4, 3, and 2 clips to RealPlayer 8. This lets you create RealPlayer animations that use Flash 4 interactive commands. If you have worked with Flash 2 in RealSystem before, note that Flash 4 commands work differently in RealPlayer. See Chapter 5 for more information.

Additional SMIL Attributes Supported in RealPlayer 8

RealPlayer 8 now supports the following SMIL 1.0 attributes, which have no effect on earlier RealPlayers:

- The `dur="indefinite"` attribute makes a clip play indefinitely, mimicking a live broadcast. For more information, see "Indefinite Durations" on page 95.
- The `repeat="indefinite"` attribute makes a clip or group loop indefinitely. For more information, see "Looping Playback Indefinitely" on page 98.
- The `fit="scroll"` attribute adds scroll bars to a clip that displays in a SMIL region that is smaller than the clip's encoded size. See "Defining How Clips Fit Regions" on page 104 for more information.

RealPlayer 8 Opens in Compact Mode

Through a Ram file, you can open RealPlayer 8 in its compact mode. See "Setting a Presentation's Starting Mode" on page 176 for instructions.

What's New in RealSystem 7?

RealSystem 7 is based on RealSystem G2 technology. It has all the capabilities of RealSystem G2, and introduces new features for RealServer 7, RealPlayer 7, and RealProducer 7. This section describes the new RealSystem 7 features that affect content authoring.

RealPlayer 7 New Features

The following features may affect how you produce streaming media for RealPlayer 7. See the RealPlayer 7 online help for information about changes to the RealPlayer 7 user interface.

PNG Images Supported

RealPlayer 7 includes support for the Portable Network Graphics (PNG) image format (`.png` file extension) in addition to JPEG and GIF formats. You can now use PNG images in both SMIL and RealPix presentations.

Pop Up Multiple RealPlayer 7 Windows

You can write hyperlinks that open content in new RealPlayer 7 windows. This lets you pop up a new RealPlayer window when a viewer clicks a link in a SMIL or RealText presentation.

Additional Information

See "Popping Up New RealPlayer Windows" on page 127.

Cache Graphics with RealPlayer 7

You can instruct RealPlayer 7 to cache image files downloaded through HTTP. This is useful for SMIL presentations that RealPlayer 7 users repeatedly view. On reloads or revisits, RealPlayer uses cached images instead of downloading the images again from the server.

Additional Information

See "Caching Files on RealPlayer" on page 132.

View SMIL Source Markup for Streamed Presentations

RealPlayer 7 has a **View>Source** command that displays the markup for the presentation's SMIL source file in your Web browser. This helps you learn how other content authors have assembled their presentations.

Additional Information

See "Viewing SMIL Source Markup" on page 205.

RealServer 7 New Features

See *RealServer Administration Guide* for full information about new features in RealServer 7. The following RealServer 7 features may affect how you produce streaming media presentations.

Ad Streaming

RealServer 7 has an advertising extension that lets you display banner ads in RealPlayer during streaming presentations. You can also stream media ads in formats such as RealVideo and Flash. This feature works with all major ad serving systems. The RealServer administrator configures most features of ad streaming.

Additional Information

For information on creating SMIL presentations that include ads, see Chapter 9.

What's New in RealSystem G2?

RealSystem G2, introduced in 1998, is a complete streaming media platform based on a new software architecture. Fundamentally different from previous versions of RealSystem, such as RealSystem 5, it uses an open architecture that allows software developers to add new features to RealPlayer and RealServer by

developing plug-ins. It is also the first streaming media system built to use a standards-based streaming protocol (RTSP) and timing language (SMIL).

New Streaming Possibilities

Text and Still Images Now Stream

RealText and RealPix let you stream text and create streaming slideshows that use special effects such as fades and zooms. In RealSystem G2, they join RealAudio, RealVideo, and Flash as standard RealSystem clip types. In addition, RealPlayer G2 and RealPlayer Plus G2 can display JPEG and GIF images.

Additional Information

See "Step 2: Choose Clip Types and Gather Tools" on page 17.

Open Plug-In Architecture Streams New File Types

RealSystem's open architecture lets RealNetworks' development partners create plug-ins to stream virtually any file type. Automatic download of plug-ins ensures that RealPlayer users can play new RealSystem streaming file types as soon as they are introduced.

Additional Information

For more on the RealNetworks developer program, visit **http://www.realnetworks.com/devzone/realdevelopers/**.

Advancements in RealAudio and RealVideo

SureStream™ Technology Encodes Single Clips for Multiple Bandwidths

The new SureStream technology available exclusively in RealSystem G2 lets you encode a single RealAudio or RealVideo clip for up to six separate bandwidths. All Web page visitors click the same link to play the clip, but their RealPlayers receive different encodings appropriate for their various connection speeds.

Additional Information

See "SureStream RealAudio and RealVideo" on page 27 for an overview of SureStream.

New RealAudio Codecs Provide Superior Sound Quality

RealSystem G2 introduces a new family of RealAudio codecs that provides fast encoding, superior sound, and the ability to encode a single clip for delivery at different bit rates using SureStream technology.

Additional Information

"RealAudio Codecs" on page 35 lists the new codecs.

New RealVideo Codec Provides Fast Encoding for Multiple Bandwidth Clips

RealSystem G2 introduces a new RealVideo codec that provides faster encoding and lets you use SureStream technology to encode a single RealVideo clip for up to six different bandwidths.

Additional Information

See "Understanding RealVideo" on page 47 for more information.

Easier Presentation Assembly

SMIL Files Coordinate Presentations

For presentations that include more than one clip, you create a SMIL file to specify how and when each clip plays. SMIL, which stands for Synchronized Multimedia Integration Language, is a standardized language that uses a simple markup similar to HTML to coordinate a streaming presentation.

Additional Information

For a look at SMIL features, see "Understanding SMIL" on page 81.

Bandwidth Negotiation Through Multiply Encoded Clips or SMIL

RealSystem G2 introduces simpler methods for supporting multiple bandwidth connections. As described previously, the new SureStream technology allows you to encode a single RealAudio or RealVideo clip for multiple connection speeds. Or you can let RealPlayer choose between different versions of a presentation based on bandwidth parameters in the SMIL file. Either way, you need just one link on your Web page, and your encoded clips do not need to conform to any naming conventions.

Additional Information

See "Reaching Multiple Audiences" on page 27 for an overview.

RealPlayer Launched Automatically

The Ramgen feature of RealServer™ can launch RealPlayer automatically. This means you do not need to create a Ram file (extension `.ram` or `.rpm`) manually. In your Web page, you link to a SMIL file or media clip, including in the URL a Ramgen parameter that causes the Web browser to launch RealPlayer and give it the SMIL file or clip.

Additional Information

See "Linking Your Web Page to RealServer" on page 168.

Enhanced Protocol Support

RTSP Now Used

Because it still supports the PNA protocol, RealServer for RealSystemG2 is backward compatible with RealSystem 3.0 through 5.0. But it introduces as its primary protocol the RealTime Streaming Protocol (RTSP), an open, standards-based protocol for multimedia streaming. Because of this, URLs that point to media clips on RealServer now begin with `rtsp://`.

RealSystem Interoperates with RTP-Based Servers and Clients

When communicating with RealPlayer, RealServer uses RealTime Streaming Protocol (RTSP) as its control protocol and RealNetworks' proprietary RDT as its packet protocol. But because RealSystem G2 also supports international standards for streaming media, RealServer and RealPlayer interoperate with RTP-based media servers and clients. The following table lists the protocols used with different mixes of servers and clients.

RealSystem Protocols

Server	Client	Control Protocol	Packet Protocol
RealServer G2 and later	RealPlayer G2 and later	RTSP	RDT
RealServer G2 and later	RTP-based client	RTSP	RTP
RTP-based server	RealPlayer G2 and later	RTSP	RTP
RealServer G2 and later	RealPlayer 3.0 to 5.0	PNA	PNA
RealServer 3.0 to 5.0	RealPlayer G2 and later	PNA	PNA

Compatibility with Previous Releases

RealSystem G2 and later are fully compatible with presentations developed for RealSystem 3.0 through 5.0. You do not need to change any existing content. RealPlayer G2 and later can play back clips streamed by an earlier version of RealServer. As well, RealServer G2 and later can stream presentations originally created for earlier versions of RealSystem. When you create a RealSystem presentation, you can use different techniques to make it compatible with earlier versions of RealPlayer.

SureStream Has a Backward-Compatibility Option

With RealSystem's SureStream technology, you can encode a single RealAudio or RealVideo clip at multiple bandwidths for RealPlayer. When you encode the clip, you select a backward-compatibility option to include an encoding for earlier versions of RealPlayer. This works only when streaming from RealServer, however, because Web servers cannot serve SureStream clips.

RealServer Can Stream Alternate Clips Through a Single URL

If you cannot use a single SureStream clip, create two clips, one for RealPlayer for RealSystem G2, and one for earlier versions of RealPlayer. You can then use RealServer's `altplay` option for Ramgen to deliver either clip through a single URL. If not using Ramgen, you can create a Ram file that lists both files.

Additional Information

See "Using Ramgen" on page 169. For information on creating a Ram file, see "Creating a Ram File Manually" on page 173.

PLANNING A STREAMING MEDIA PRESENTATION

A streaming presentation can consist of one or many clips. No matter how simple or complicated your presentation, you'll need to plan your media production so you can work effectively and reach your target audience. This chapter explains the basics of how to put streaming media presentations together. If you are not yet familiar with RealSystem components, see also Appendix A on page 189.

Step 1: Decide How to Deliver Clips

The first step in creating a streaming presentation is to consider the last step: how will you deliver your clips to other people? How you plan to stream your clips can greatly affect your media production.

RealServer Streaming

RealServer is the primary vehicle for streaming clips to RealPlayer. It delivers clips at many bandwidths, ensuring that streamed presentations stay synchronized. A RealServer administrator sets up and runs each RealServer. If you will not be running RealServer yourself, check the following with your RealServer administrator:

1. What version of RealServer is available?

   To deliver clips described in this manual, you'll need RealServer G2, 7, or 8, rather than any of the earlier versions. Make sure that your RealServer can deliver all the clips you plan to develop.

2. How many streams can RealServer serve?

   Each RealServer has a maximum number of media streams it can send out at once. A RealServer with a maximum of 500 streams, for example, can stream video to 500 viewers simultaneously. Make sure that the RealServer you plan to use has adequate capacity for your needs.

3. Are there any bandwidth constraints?

   The RealServer computer may lack the outgoing bandwidth to deliver a lot of high-speed clips simultaneously. If you plan to develop high-bandwidth presentations, confer with the RealServer administrator about bandwidth limitations.

4. Where will your clips reside?

   Your clips typically reside on RealServer, whereas your Web pages are on a Web server. You'll need to know the URLs for your clips on RealServer so that you can set up your Web page hyperlinks correctly.

5. Do any RealServer features need to be set up?

   The RealServer administrator can set up many streaming and security features, such as:

   - Live broadcasts
   - Pay-per-view content
   - Automatic ad insertion
   - Password authentication

Using RealServer through an Internet Service Provider

If an Internet service provider (ISP) hosts your Web pages, contact the ISP administrator to check out the RealServer issues described above. Also find out how much disk space you will have for streaming media. Many ISPs allot you a certain amount of disk space on their servers, such as 5 or 10 MB. Although this is a generous amount for Web pages, it's not much for streaming media. A single video clip can easily take up that much space.

Web Server Downloading

Although Web servers can deliver some streaming clips, they don't have RealServer's ability to synchronize clips and keep long presentations flowing smoothly. When only a Web server is available, you can still deliver multimedia presentations, but you will not be able to use all of the features that RealSystem offers.

Additional Information

If you plan to deliver clips with a Web server, read "Limitations on Web Server Playback" on page 171.

Local Playback

You can also create presentations that play back from a user's local computer. An example of this is a multimedia-enhanced book written with HTML and containing links to RealSystem clips. Users download the files to their computers, playing back the media clips with RealPlayer. In this case, you produce clips as described in this production guide, except that you don't target specific network connection bandwidths. In the HTML pages, URLs point to clips on the user's computer instead of on RealServer.

Additional Information

For more on local URLs in SMIL files, see "Linking to Local Clips" on page 88. See also "Creating a Ram File Manually" on page 173.

Step 2: Choose Clip Types and Gather Tools

RealSystem gives you many possibilities for creating streaming media. Your presentation may consist of a single clip, or several clips that play together. As you decide what types of clips you want to stream, gather the production tools you'll need to make the clips.

Tip

The HTML version of this production guide, available at **http://service.real.com/help/library/encoders.html**, contains clip samples you can view with RealPlayer. Visit **http://www.realstore.com** to find out about the many tools available to help you create streaming media.

Audio and Video

RealAudio and RealVideo are the most popular streaming media formats. To produce them, you run an encoding tool with audio or video input from a live source, a recording format such as tape or CD, or a digitized file in a standard format such as WAV, AVI, QuickTime, or MPEG.

Audio and Video Production Tools

A streaming RealAudio or RealVideo clip results from gathering, editing, and encoding audio or video input. To carry out the initial steps of gathering and editing content, you'll need the following:

- A video camera and a microphone

  To capture live input, use any video camera and microphone that can be attached to your computer. You will not need these devices, though, if your audio or video source is already digitized.

- An audio/video capture card

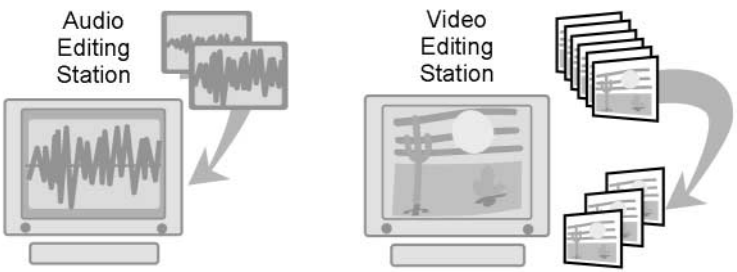
  To take input from a microphone or camera, your computer needs an audio/video capture card. This card accepts the input and digitizes it into a format you can edit. On Windows computers, you can use any video capture card that supports Video for Windows.

- Audio and video editing software

  These programs let you edit digitized audio and video files. When creating clips that stream on demand, it's best to edit and optimize the input before encoding it. When broadcasting, you can convert audio and video input into RealAudio or RealVideo clips directly from a capture card without first creating a separate, digitized file.

Producing RealAudio and RealVideo does not require that you use specific microphones, cameras, capture cards, or editing tools. Just ensure that your editing tools can save files in formats you can easily convert to streaming formats with your encoding tool.

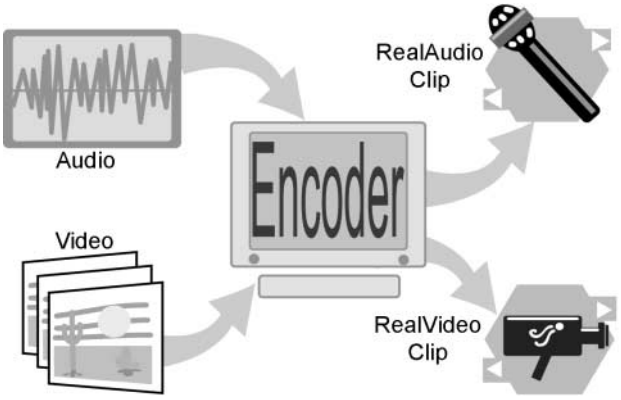
Use Your Favorite Editing Tools to Prepare Audio and Video Files



RealAudio and RealVideo Encoding Tools

Some editing programs can export digitized audio and video directly to RealAudio or RealVideo. If your editing program cannot export clips or you don't want to use this feature, you can use a RealNetworks tool to encode clips from files in standard formats such as WAV, AVI, QuickTime, and MPEG. RealProducer Basic is a free tool for encoding RealAudio and RealVideo clips. RealProducer Plus is an enhanced version that offers more encoding features.

RealProducer Creates Streaming Clips



Additional Information

For more information about RealProducer, see "Getting Production Tools" on page 190. You can find tools at **http://www.realnetworks.com/products/index.html**.

SMIL

When you want to combine two or more clips into a single presentation, you use SMIL. Pronounced "smile," SMIL is a simple markup language that tells RealPlayer how to lay out and play your clips. You can use any word processor or text editor to write SMIL. For basic information about SMIL, see "Writing SMIL Files" on page 192. For details, see Chapter 6.

Animation

With Macromedia Flash, you can build anything from streaming cartoons to e-commerce applications. To create a soundtrack, you can encode a RealAudio clip that streams along with the Flash clip. See Chapter 5 for details about producing Flash animation for RealPlayer. Learn more about Flash from Macromedia's Web site at:

http://www.macromedia.com/software/flash/

Images

Streaming multimedia presentations played with RealPlayer can include still images in the following formats:

- GIF87, GIF89, and animated GIF (.gif)

  Both interlaced and noninterlaced GIFs will work, but noninterlaced GIFs are recommended.

- JPEG (.jpg)

  RealPlayer can display RGB baseline JPEGs. Progressive and grayscale JPEGs are not supported.

- PNG (.png)

  RealPlayer does not adhere to gamma settings in PNG images.

Note

Image transparency is not supported between SMIL regions. For more on this, see "Transparency in SMIL Regions" on page 104.

Images in SMIL Presentations

To add images to streaming presentations as backgrounds or buttons, for example, simply incorporate the images by using SMIL. This way, you can specify exactly where images appear in relation to your clips. You can also use SMIL to turn images into hyperlinks.

RealSlideshow Presentations

When you want to create a streaming slideshow, the easiest solution is to use RealSlideshow™ or RealSlideshow Plus. These tools have drag-and-drop interfaces that let you quickly build your slideshow, which can include text captions, audio narrations, and background music. RealNetworks' partners will even host your streaming slideshow. Get RealSlideshow at:

http://www.realnetworks.com/products/index.html

RealPix Markup

Streaming slideshows are based on the RealPix markup language. Instead of using RealSlideshow, you can write your own markup to assemble images into a RealPix presentation that has eye-catching special effects such as dissolves and zooms. Learn the markup from *RealPix Authoring Guide*:

http://service.real.com/help/library/encoders.html

Text

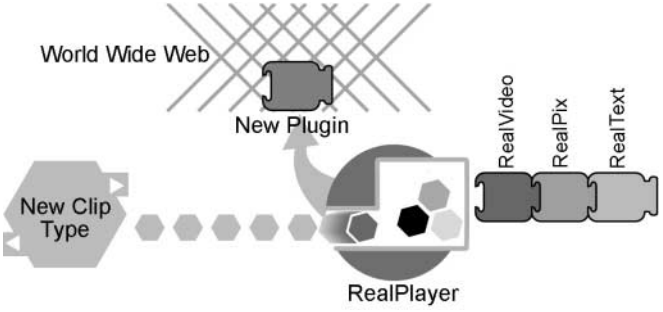
Unlike HTML, SMIL does not display text directly. To show text in RealPlayer, you can add text to any image, video, or animation clip. Or you can use RealText, which streams text at specific times within a presentation. RealText lets you subtitle videos, for example, or create hypertext links. Learn the RealText markup from *RealText Authoring Guide*:

http://service.real.com/help/library/encoders.html

RealPlayer's Update Features

RealPlayer's plug-in and autoupdate technologies ensure that your clips can reach the widest audience possible. RealPlayer plug-ins function like Web browser plug-ins. If RealPlayer doesn't have a plug-in needed to play a particular streaming clip, it downloads that plug-in from the Internet. RealPlayer can even use its autoupdate technology to upgrade itself to a new version when necessary.

RealPlayer Downloads Plug-ins it Needs from the Internet



Tip

RealPlayer's plug-in technology lets it play many types of clips developed by RealNetworks' partners. Check **http://www.realnetworks.com/devzone** for news about other types of streaming media.

Compatibility with Earlier Versions of RealPlayer

Plug-in and autoupdate technologies were introduced with RealPlayer G2. Earlier versions of RealPlayer cannot upgrade themselves, so they cannot play all the clips described in this production guide. Generally, you don't need to be concerned with backward compatibility because most RealPlayer users

upgrade to the latest release. The following table summarizes which versions of RealPlayer offer which features. RealPlayer 4.0, for example, plays only RealAudio and RealVideo.

RealPlayer Features, from the Latest Version of RealPlayer to the Earliest

Feature	8	7	G2	5	4	3	2	1
RealAudio streaming	X	X	X	X	X	X	X	X
RealVideo streaming	X	X	X	X	X	–	–	–
Flash 2.0 streaming	X	X	X	X	–	–	–	–
Flash 3.0 and 4.0 streaming	X	–	–	–	–	–	–	–
RealPix streaming	X	X	X	–	–	–	–	–
RealText streaming	X	X	X	–	–	–	–	–
SMIL presentations	X	X	X	–	–	–	–	–
Plug-ins for additional clip types	X	X	X	–	–	–	–	–
Autoupdate	X	X	X	–	–	–	–	–

This table covers only general clip compatibility, not codec compatibility. The latest version of RealProducer creates RealAudio and RealVideo clips with codecs that RealPlayer 4 and earlier versions cannot play back. RealProducer has a compatibility option for RealPlayer 5, making that the earliest version of RealPlayer you can reach with current production methods.

Protection of Copyrighted Content

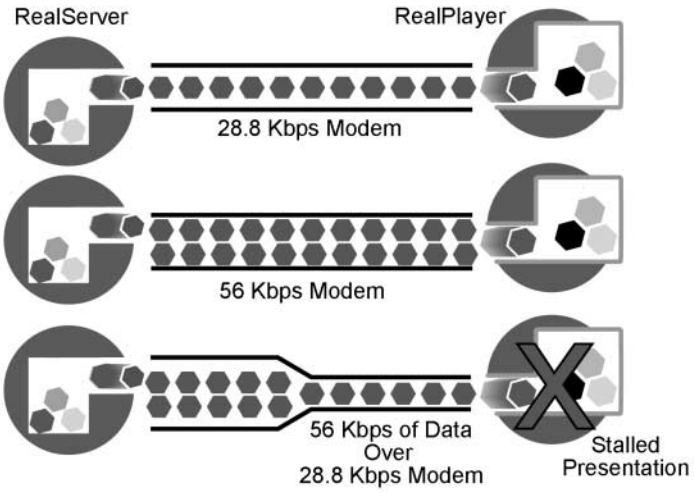
Unlike a Web browser, RealPlayer does not store clips in a disk cache or allow users to copy or download still images. This helps you keep copyrighted material secure when you stream clips from RealServer, though not from a Web server. A RealProducer option lets you encode RealAudio and RealVideo clips so that viewers can record the clips on their computers.

Step 3: Develop a Bandwidth Strategy

Any computer connected to a network has a connection bandwidth, which is a maximum speed at which it can receive data. Web users with 28.8 Kbps modems, for example, can view only those presentations that stream less than 28.8 Kb of data per second. Presentations that stream more data than that per second may stall because the data cannot get over the modems fast enough to keep the clips flowing. These presentations will not cause problems for users with faster connections, though.

Successfully targeting your audience's connection bandwidth is crucial for developing streaming media. Viewers don't like to wait more than a few seconds for playback to begin after they click a link. And if your clips sputter because they use too much bandwidth, viewers are not likely to stay tuned. Developing a bandwidth strategy helps ensure that clips play back quickly and don't stall. You can also devise ways to deliver good clips to users with slow connections, and great clips to those with fast connections.

Presentation Data Must Fit RealPlayer's Bandwidth



Buffering

For each streaming clip, RealPlayer keeps a "buffer" that acts as a data reservoir. Data enters the buffer as it streams to RealPlayer, leaving the buffer as RealPlayer plays the clip. The buffer helps ensure that lapses of available bandwidth don't stall the presentation. If network congestion halts the flow of data for a few seconds, for example, RealPlayer keeps the clip playing with the buffered data. Your goal is to minimize initial buffering and eliminate rebuffering.

Initial Buffering (Preroll)

RealPlayer buffers a few seconds of data before a clip plays. Also called "preroll," initial buffering is required for every clip. Developing clips that use an appropriate amount of bandwidth keeps preroll to an acceptable level. You want preroll to be low—less than 15 seconds for each clip. RealAudio and

RealVideo encoding tools set a low preroll for you. With other clips, though, how you create the clip determines its preroll.

Tip

By using SMIL, you can mask the preroll between clips. Once you're familiar with SMIL, refer to "Smoothing Transitions Between Clips" on page 205.

Rebuffering

When clip data has stopped coming in and the clip buffer is empty, RealPlayer has to halt clip playback to store data again, or "rebuffer." Sometimes this is unavoidable because the viewer's available bandwidth drops for too long. When developing a multiclip presentation, though, you need to consider timelines carefully so that you don't inadvertently cause rebuffering, which can happen if too many clips fight for too little bandwidth.

Audience Bandwidth Targets

Your streaming presentations should never consume all of your audience's connection bandwidth. They must always leave bandwidth for network overhead, error correction, resending lost data, and so on. Otherwise, they may require frequent rebuffering. The following table recommends maximum streaming speeds for common network connections. To reach 28.8 Kbps modems, for example, a presentation should stream no more than 20 Kb of data per second.

Maximum Streaming Rates

Target Audience	Maximum Streaming Rate
14.4 Kbps modem	10 Kbps
28.8 Kbps modem	20 Kbps
56 Kbps modem	34 Kbps
64 Kbps ISDN	45 Kbps
112 Kbps dual ISDN	80 Kbps
Corporate LAN	150 Kbps
256 Kbps DSL/cable modem	225 Kbps
384 Kbps DSL/cable modem	350 Kbps
512 Kbps DSL/cable modem	450 Kbps

For any other connection speed, calculate the maximum streaming speed as:

- Approximately 75 percent of the connection bandwidth for analog connections such as dial-up modems.

  –Or–

- Approximately 90 percent of the connection bandwidth for high-speed digital connections such as DSL or cable modems.

Multiclip Presentations

When several clips are played together, their streaming speeds added together should not exceed the connection maximum. For example, RealPix and RealAudio clips streaming at 12 and 8 Kbps, respectively, can play in parallel over 28.8 Kbps modems because together they stream at 20 Kbps. However, they cannot play back together if they stream at 12 and 16 Kbps, respectively, because the 28 Kbps total streaming speed leaves the modem no bandwidth for overhead. Such a presentation would likely require frequent rebuffering.

Streaming at Less than the Maximum Speed

Your presentations do not have to stream at the maximum speeds listed in the preceding table. In some cases, you may want your clips to stream at less than the maximum:

- You may need to leave enough bandwidth for the user to perform other network activities. When streaming an Internet radio station, for example, leave some bandwidth for the listener to view Web pages.
- Bandwidth is shared by everyone on a local area network (LAN). If the LAN is heavily used, the 150 Kbps LAN target speed may slow down the LAN too much. For an intranet, the LAN manager should decide the maximum streaming rate.

Clip Bandwidth Characteristics

To reach your target audiences with your clips, you need to understand your clips' bandwidth characteristics.

RealAudio and RealVideo

A RealAudio and RealVideo encoding tool can turn your source audio or video file into a clip that streams to any target connection with little preroll. But if the tool has to squeeze a file down too much to reach a low-bandwidth target, clip quality may degrade. So although the clip will stream well, you might not like the results. To ensure good-quality playback, keep your streaming

bandwidths in mind when creating source files, especially when you plan to reach dial-up modem users.

Additional Information

See "Understanding RealAudio" on page 33 and "Understanding RealVideo" on page 47.

Flash

Macromedia Flash streams well at low bandwidths, making it an attractive alternative to video. Low streaming speed doesn't affect Flash's visual quality as it can with video. At low bandwidths, though, you may not be able to include as many items in your animated scenes as when streaming at high bandwidths. After you develop a Flash clip for RealPlayer, you tune it to stream at a specific bit rate. For more on this, see "Flash Bandwidth Characteristics" on page 66.

RealText and SMIL

Because RealText and SMIL files are plain text, they use little bandwidth. You generally don't need to be concerned about how they affect a presentation's bandwidth consumption.

RealPix (Slideshows)

RealPix bandwidth use depends on the image sizes and how soon each image must appear in the clip's timeline. At higher bandwidths, you can use larger images and display them at shorter intervals. By varying image size and the RealPix timeline, you gain a lot of control over bandwidth use. RealSlideshow always ensures that images stream at a rate appropriate for your target audience. If you write RealPix markup by hand, though, you need to be careful not to overload a connection's bandwidth.

Images in SMIL Presentations

JPEG, GIF, or PNG images in a SMIL presentation stream at 12 Kbps. See "Defining Image Options" on page 123 for instructions on changing this streaming bit rate.

Reaching Multiple Audiences

To provide good content for users with slower connections, and great content for those with faster connections, you can use two methods, combining them if needed:

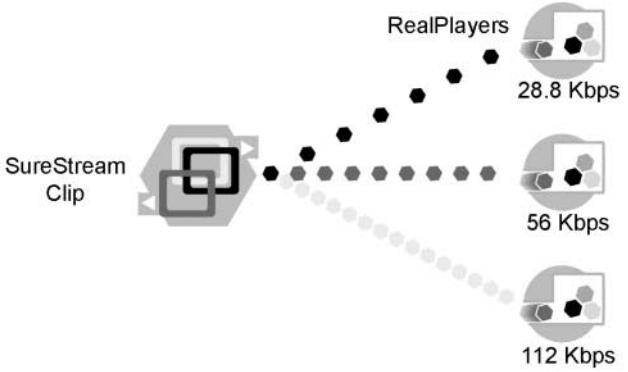
- Create a single RealAudio or RealVideo clip that targets different audience bandwidths by using SureStream technology.
- Create separate clips for each bandwidth target, and let RealPlayer choose which set of clips to play through SMIL.

Either way, you add to your Web page just one link for all visitors. You don't need separate links for modems and DSL connections, for example.

SureStream RealAudio and RealVideo

With RealSystem's SureStream technology, you can encode a RealAudio or RealVideo clip for multiple bandwidths. For example, you can encode a single RealAudio music clip for 28.8 Kbps modems, 56 Kbps modems, 112 Kbps dual ISDN, 256 Kbps DSL, and so on. The clip's playback quality improves with each faster speed. When a viewer clicks a link to a SureStream clip, RealPlayer and RealServer determine which stream to use based on the available bandwidth, as shown in the following illustration.

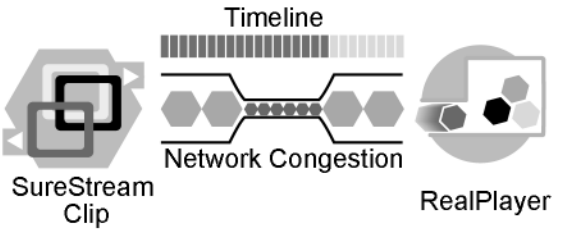
SureStream Clip Encoded for Multiple Bandwidths



RealServer and RealPlayer can even adjust this choice to compensate for network conditions. If a fast connection becomes bogged down because of high network traffic, RealServer switches to a lower-bandwidth stream to prevent the presentation from stalling. When the congestion clears, RealServer

switches back to the higher-bandwidth stream. RealPlayer doesn't need to rebuffer data during this shifting.

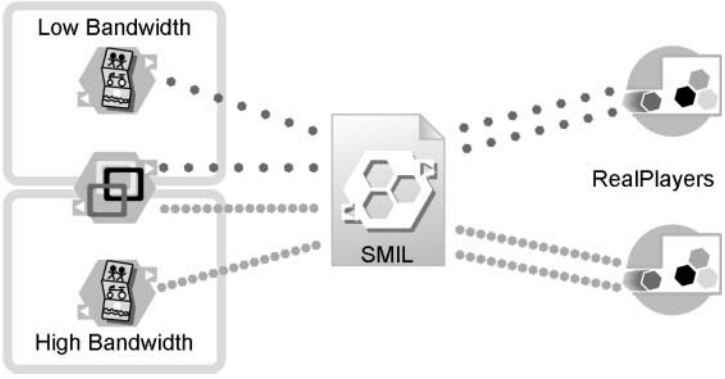
Switching Bandwidths During Network Congestion



Switching Between Multiple Clips with SMIL

Only RealAudio and RealVideo clips can stream at multiple bandwidths. You can create multiple versions of other clips, though, each for a different bandwidth. RealPlayer then chooses which clip to play based on a SMIL bandwidth parameter. The following illustration shows a SMIL file that lists separate high-bandwidth and low-bandwidth RealPix clips. Each RealPlayer evaluates the SMIL file and chooses the RealPix clip appropriate for its connection speed. Both presentations use the same SureStream RealAudio clip, though, which has been encoded internally for multiple bandwidths.

Bandwidth Choices through a SureStream Clip and SMIL



When you use SMIL for bandwidth choices, RealServer cannot downshift to a lower-bandwidth clip group the way it can downshift to a slower SureStream stream. RealServer employs other techniques, though, to compensate for network congestion. Its stream thinning capabilities enable it to drop low-priority data to decrease the presentation bandwidth temporarily. When the

network congestion clears, RealServer continues to stream all the presentation data.

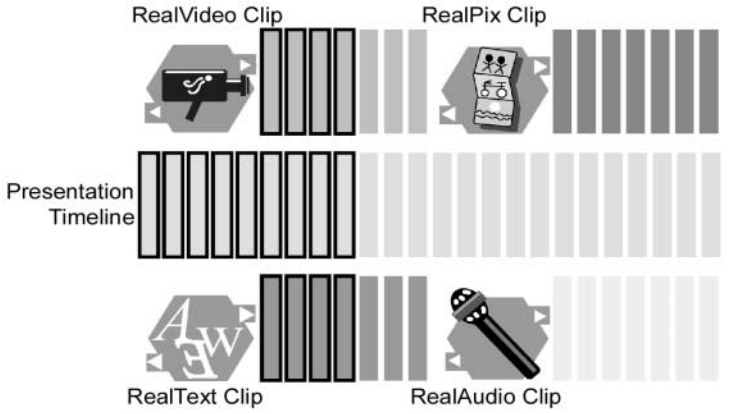
Additional Information

"Setting Bandwidth Choices" on page 110 explains how to use SMIL to designate different bandwidth groups.

Step 4: Organize the Presentation Timeline

Every streaming media clip has a timeline. A RealAudio clip may play for five minutes, for example, giving it a five-minute timeline. When clips are streamed together, you have a presentation timeline as well. Before producing clips, plan the presentation timeline. Among other things, the timeline can determine the order in which you produce clips. A well-conceived timeline also helps ensure that clips do not overload a connection's bandwidth and cause rebuffering.

Clip Timelines Coordinate with a Presentation Timeline



Timeline Considerations

When you assemble a streaming media presentation, you can manipulate various aspects of clip timelines.

Clips with Internal Timelines

Audio, video, and animation have internal timelines. In a 10-minute video, for instance, each frame corresponds to a specific point in a 10-minute timeline. Each second of audio meshes with each second of the image throughout the

clip's overall timeline. Your video, audio, or animation software is your main tool for manipulating the clip's timeline, which is woven into the fabric of the clip.

Clips with Variable Timelines

With RealPix or RealText, you use the markup language to control when each image or text block appears and how long it lasts. When combining clips, it's typically easier to produce audio, video, or animation first. Then set the RealPix and RealText timelines to coordinate with those clips.

SMIL Timing Commands

A SMIL file can include its own timing elements. Timing a presentation with SMIL can be as simple as having one clip start as soon as another one stops. But you can also use commands to delay playback for 10 seconds, for example, or to have a clip start playing 30 seconds into its internal timeline. SMIL's timing commands are optional, but they give you the flexibility you may need for some presentations.

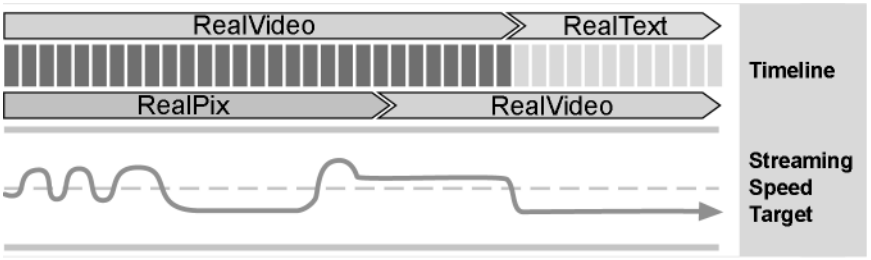
Additional Information

For more on SMIL timing, see "Specifying Timing" on page 92.

Timelines for Multiclip Presentations

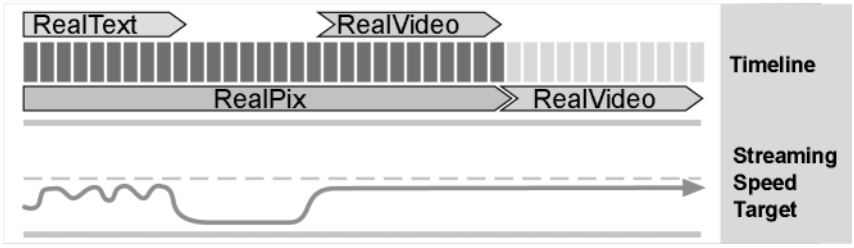
For presentations that include multiple clips, consider how to group clips without overloading an audience's connection bandwidth. The following illustration shows poor timeline planning. At various points, RealVideo and RealPix clips playing together exceed the connection's maximum streaming speed, which is represented below by the dashed line. Illustrated by the solid line, bandwidth use peaks again when the second RealVideo clip begins to play before the first video clip finishes. This presentation requires a high preroll for clips and would likely result in rebuffering at peak points.

Poor Bandwidth Use in a Multiclip Presentation



The next illustration shows better timeline planning and bandwidth management. The presentation starts with a low-bandwidth RealText clip that does not interfere with the streaming of the images in the RealPix clip. A RealVideo clip starts after the RealPix clip has streamed all of its images and does not need any more bandwidth. The second RealVideo clip starts after the first RealVideo clip has ended, so the two clips do not compete for bandwidth.

Improved Bandwidth Use in a Multiclip Presentation



Timeline Management

When developing a streaming presentation, keep the following in mind:

- Consider the presentation timeline carefully to eliminate bandwidth bottlenecks. These typically occur when two or more high-bandwidth clips play simultaneously. You may need to omit high-bandwidth pairings, combining high-bandwidth clips with low-bandwidth clips instead.
- Stagger the start times for clips. Every clip requires a certain amount of preroll before RealPlayer can play it. Your presentation will flow more smoothly if RealServer does not need to send more than one clip's preroll at a time.
- Start presentations with low-bandwidth clips. For example, use RealText to display credits. Or begin with a highly compressed RealAudio narration

before bringing in any other clips. RealSystem can take advantage of the extra bandwidth to begin streaming higher-bandwidth data to RealPlayer "behind the scenes."

Additional Information

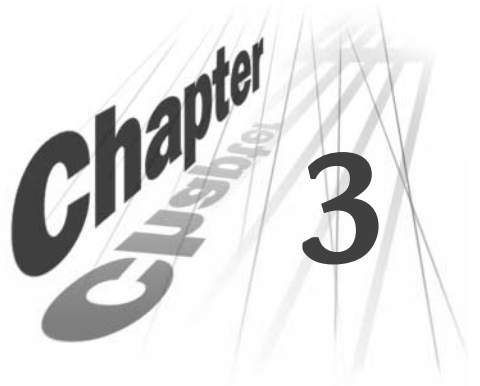
See "Smoothing Transitions Between Clips" on page 205.

- Test your presentations in "real world" circumstances, replicating your audience's bandwidth conditions. Clips may play back OK from your desktop computer but bog down when streamed over a modem.

Step 5: Create Your Clips

When you've decided how you'll stream clips, chosen clip types and tools, developed a bandwidth strategy, and planned a timeline, you're ready to start creating streaming presentations. The rest of this guide provides production pointers, but does not explain how to use any specific tools. Be sure to have the documentation for your production tools handy as you develop your clips.

- If you're producing audio, video, or animation, read Chapter 3, Chapter 4, or Chapter 5, respectively.
- For information on SMIL, refer to Chapter 6, Chapter 7, and Appendix D.
- To add clips to a Web page, follow the instructions in Chapter 8.
- When you're ready to move clips to servers, read Chapter 10.
- If you plan to broadcast presentations live, see Chapter 11.



PRODUCING AUDIO

RealNetworks pioneered streaming audio with RealAudio, the first streaming media product for the Internet. Since its debut in 1995, RealAudio has become the standard for network audio, delivering stereo sound over 28.8 Kbps modems and CD-quality sound at high connection speeds. This chapter gives pointers on how to prepare and encode your sound files for streaming.

Understanding RealAudio

Because RealAudio clips are compressed, you typically start with a sound file in a digitized, uncompressed format such as WAV or AIFF. Using a RealAudio encoding tool, you create a RealAudio clip from the source file. RealAudio clips use the file extension `.rm`, although older clips may use `.ra` instead. This section explains how RealAudio encodes an audio file for streaming. This knowledge will help you produce high-quality streaming clips.

Bandwidth and Audio Quality

One way that RealAudio squeezes an audio file's size down is by throwing out nonessential data. This makes it a *lossy* compression format. RealAudio doesn't delete data indiscriminately, though. It first jettisons portions you cannot hear, such as very high and very low frequencies. Next, it removes as much data as needed while keeping certain frequencies intact. Voice encoding favors frequencies in the normal human speaking range. Music encoding retains a broader frequency range.

Although RealAudio is savvy about what audio data it throws out, be aware that the lower the connection speed, the more data gets ejected, and the cruder the sound quality becomes. At low bandwidths, you get roughly the quality of an AM radio broadcast. With faster connections, you can encode music with FM-quality sound. And at the high speeds of DSL, cable modems, and LANs, RealAudio sound quality rivals that of CD playback. When creating

RealAudio clips for low bandwidths, it's important to start with high-quality input, as described in "Capturing Audio" on page 41, to attain good sound quality.

RealAudio Bandwidth Characteristics

You create a RealAudio clip by using one or more RealAudio *codecs*. A codec is a coder/decoder. It tells an encoding tool how to turn audio source files into RealAudio clips. On the receiving end, RealPlayer uses codecs to expand clips into audio data the computer can play. RealAudio employs a series of codecs, each of which creates an audio stream for a precise bandwidth. One codec compresses mono music for a 28.8 Kbps modem. Another one compresses stereo music for that same modem speed. This set of codecs is different from the set used to compress music for, say, DSL and cable modem connections.

A RealAudio clip consumes bandwidth at a flat rate determined by the codec used to encode the clip. A RealAudio clip encoded with a 20 Kbps codec, for example, steadily consumes 20 Kbps of bandwidth as it plays. The following table lists the standard bit rates for RealAudio clips encoded for specific target audiences by RealProducer 8.5. Encoding a voice-only audio file for a 28.8 Kbps modem, for example, creates a 16 Kbps streaming clip. With mono music input, though, you get a 20 Kbps clip.

RealAudio Standard Bit Rates

<table>
<tr><th>Target Audience</th><th>Voice Only</th><th>Voice and Music</th><th>Mono Music</th><th>Stereo Music</th></tr>
<tr><td>28.8 Kbps modem</td><td rowspan="2">16 Kbps</td><td>20 Kbps</td><td>20 Kbps</td><td>20 Kbps</td></tr>
<tr><td>56 Kbps modem</td><td>32 Kbps</td><td>32 Kbps</td><td>32 Kbps</td></tr>
<tr><td>64 Kbps single ISDN</td><td>32 Kbps</td><td>44 Kbps</td><td>44 Kbps</td><td>44 Kbps</td></tr>
<tr><td>112 Kbps dual ISDN</td><td rowspan="3">64 Kbps</td><td>64 Kbps</td><td rowspan="4">64 Kbps</td><td>64 Kbps</td></tr>
<tr><td>Corporate LAN</td><td rowspan="3">96 Kbps</td><td>132 Kbps</td></tr>
<tr><td>256 Kbps DSL/cable modem</td><td>176 Kbps</td></tr>
<tr><td>384 Kbps DSL/cable modem</td><td rowspan="2">96 Kbps</td><td>264 Kbps</td></tr>
<tr><td>512 Kbps DSL/cable modem</td><td>352 Kbps</td></tr>
</table>

In terms of bandwidth use, RealAudio is the most *in*flexible media type. The RealAudio codecs set streaming bit rates in a stairstep model: 20 Kbps, 36 Kbps, 44 Kbps, and so on, with no inbetween choices. Because RealAudio clips always stream at specific bit rates, consider their bandwidth needs first when

you use them in multiclip presentations. Then create your other clips to stream within the bandwidth that's left.

Note

With SureStream technology, a single RealAudio clip can stream at many different speed. For the basics of SureStream, see "SureStream RealAudio and RealVideo" on page 27.

RealAudio Codecs

This section discusses the RealAudio codecs used by RealProducer. The codecs are listed in separate tables for voice, mono music, and stereo music. Voice codecs focus on the standard frequency range of the human voice. Music codecs have broader frequency responses to capture more high and low frequencies. The tables list each codec's optimum sampling rate and frequency response.

- Sampling rate

  Using a codec's optimum sampling rate in your audio source file ensures that the audio stays synchronized with other media in the presentation. It also prevents pitch shifting in audio resampling. Audio quality degrades if you use less than the optimum sampling rate. If you use a higher rate, it is best to use a multiple of the optimum rate. If the optimum rate is 8 kHz, for example, use a higher rate of 16 kHz or 32 kHz. When in doubt, use the CD-quality sampling rate of 44.1 kHz.

  Note

  RealProducer 8.5 and later can also take audio input sampled at a rate of 48 kHz, which is commonly used with digital video discs (DVD) and digital audio tape (DAT). RealProducer automatically resamples the input to the optimum rates for the selected RealAudio codecs.

- Frequency response

  Each codec has a particular frequency response measured in kilohertz (kHz). A codec with a higher frequency response reproduces a wider range of sound than a codec with a lower response. A measure of codec quality, the frequency response does not affect how you produce audio source files. RealAudio encoding always results in a clip of equal or lower quality

than the original audio source. If the original audio source has an 8 kHz frequency response, encoding it with a codec that has a frequency response of 10 kHz produces a clip that still has a response of 8 kHz.

Voice Codecs

Voice codecs are for voice-only clips. The lowest-speed voice codec normally used to encode a RealAudio clip streams data at 16 Kbps. The lower-speed codecs (5, 6.5, and 8.5 Kbps) are used as SureStream duress streams that RealPlayer downshifts to if the connection bandwidth drops. They're also used to encode soundtracks for low-bandwidth RealVideo clips.

RealAudio Voice Codecs

RealAudio Codec	Sampling Rate	Frequency Response
5 Kbps Voice	8 kHz	4 kHz
6.5 Kbps Voice	8 kHz	4 kHz
8.5 Kbps Voice	8 kHz	4 kHz
16 Kbps Voice	16 kHz	8 kHz
32 Kbps Voice	22.05 kHz	11 kHz
64 Kbps Voice	44.1 kHz	20 kHz

Mono Music Codecs

As with the voice codecs, the lowest-speed mono music codec normally used with RealAudio streams data at 16 Kbps. The lower-speed codecs (6, 8, and 11 Kbps) are used as duress streams in SureStream clips, and to encode soundtracks for low-bandwidth RealVideo clips. When there are two versions of a codec, RealProducer uses the higher-response version by default.

RealAudio Mono Music Codecs

RealAudio Codec	Sampling Rate	Frequency Response
6 Kbps Music	8 kHz	3 kHz
8 Kbps Music	8 kHz	4 kHz
11 Kbps Music	11.025 kHz	5.5 kHz
16 Kbps Music	22.05 kHz	8 kHz
20 Kbps Music	22.05 kHz	10 kHz
20 Kbps Music—High Response	44.1 kHz	14.5 kHz
32 Kbps Music	44.1 kHz	14.5 kHz

(Table Page 1 of 2)

RealAudio Mono Music Codecs (continued)

RealAudio Codec	Sampling Rate	Frequency Response
32 Kbps Music—High Response	44.1 kHz	16 kHz
44 Kbps Music	44.1 kHz	20 kHz
64 Kbps Music	44.1 kHz	20 kHz

(Table Page 2 of 2)

Stereo Music Codecs

RealPlayer G2 or later can play the stereo codecs listed in the following table. These stereo codecs do not stream slower than 20 Kbps because, if they did, they would not have enough frequency response for adequate sound.

RealAudio Stereo Music Codecs

RealAudio Codec	Sampling Rate	Frequency Response
20 Kbps Stereo Music	11.025 kHz	5 kHz
32 Kbps Stereo Music	22.05 kHz	8 kHz
44 Kbps Stereo Music	22.05 kHz	11 kHz
64 Kbps Stereo Music	44.1 kHz	16 kHz
96 Kbps Stereo Music	44.1 kHz	16 kHz

RealAudio 8 Stereo Music Codecs

RealAudio 8 introduces new stereo music codecs that greatly enhance sound quality, especially over high-speed connections. Only RealPlayer 8 and later can play RealAudio 8 clips. Users with RealPlayer G2 and RealPlayer 7 are prompted to upgrade to the latest version of RealPlayer before listening to a RealAudio 8 clip. When there are two versions of a codec, RealProducer uses the higher-response version by default. These codecs are also used to encode voice-with-music clips.

RealAudio 8 Stereo Music Codec s

RealAudio 8 Codec	Sampling Rate	Frequency Response
16 Kbps Stereo Music	22.05 kHz	4.3 kHz
20 Kbps Stereo Music	22.05 kHz	8.6 kHz
20 Kbps Stereo Music—High Response	22.05 kHz	9.9 kHz
32 Kbps Stereo Music	22.05 kHz	10.3 kHz
32 Kbps Stereo Music—High Response	44.1 kHz	13.8 kHz

(Table Page 1 of 2)

RealAudio 8 Stereo Music Codecs (continued)

RealAudio 8 Codec	Sampling Rate	Frequency Response
44 Kbps Stereo Music	44.1 kHz	13.8 kHz
44 Kbps Stereo Music—High Response	44.1 kHz	16.0 kHz
64 Kbps Stereo Music	44.1 kHz	16.0 kHz
96 Kbps Stereo Music	44.1 kHz	16.0 kHz
105 Kbps Stereo Music	44.1 kHz	13.7 kHz
132 Kbps Stereo Music	44.1 kHz	16.5 kHz
146 Kbps Stereo Music	44.1 kHz	16.5 kHz
176 Kbps Stereo Music	44.1 kHz	19.2 kHz
264 Kbps Stereo Music	44.1 kHz	22.0 kHz
352 Kbps Stereo Music	44.1 kHz	22.0 kHz

(Table Page 2 of 2)

Older RealAudio Codecs

The following tables list older RealAudio codecs that became obsolete with RealSystem G2. Some of these codecs are still used for compatibility with RealPlayer 5, though. RealPlayer G2 and later can play any clips encoded with these codecs. An "X" in the 5, 4, 3, 2, or 1 column indicates that a clip encoded with this codec can be played by that version of RealPlayer.

Older RealAudio Voice Codecs

RealAudio Codec	5	4	3	2	1	Sampling Rate	Frequency Response
8 Kbps Voice	X	X	X	X	X	8 kHz	4 kHz
15.2 Kbps Voice	X	X	X	X	–	8 kHz	4 kHz

Older RealAudio Mono Music Codecs

RealAudio Codec	5	4	3	2	1	Sampling Rate	Frequency Response
8 Kbps Music	X	X	–	–	–	8 kHz	4 kHz
12 Kbps Music	X	X	–	–	–	8 kHz	4 kHz
16 Kbps Music Low Response	X	X	X	–	–	8 kHz	4 kHz
16 Kbps Music Medium Response	X	X	X	–	–	11.025 kHz	4.7 kHz
16 Kbps Music High Response	X	X	X	–	–	11.025 kHz	5.5 kHz
32 Kbps Music	X	X	–	–	–	16 kHz	8 kHz

(Table Page 1 of 2)

Older RealAudio Mono Music Codecs (continued)

RealAudio Codec	5	4	3	2	1	Sampling Rate	Frequency Response
40 Kbps Music	X	X	X	–	–	22.05 kHz	11 kHz
80 Kbps Music	X	X	X	–	–	44.1 kHz	20 kHz

(Table Page 2 of 2)

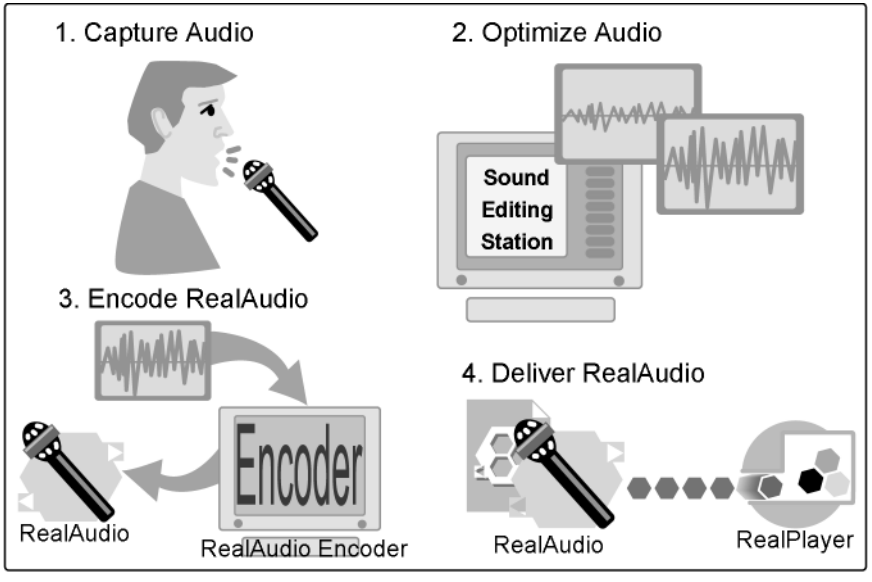
Older RealAudio Stereo Music Codecs

RealAudio Codec	5	4	3	2	1	Sampling Rate	Frequency Response
20 Kbps Stereo Music	X	X	X	–	–	8 kHz	4 kHz
32 Kbps Stereo Music	X	X	–	–	–	11.025 kHz	5.5 kHz
40 Kbps Stereo Music	X	X	X	–	–	16 kHz	8 kHz
80 Kbps Stereo Music	X	X	X	–	–	32 kHz	16 kHz

Steps for Streaming RealAudio

To produce a great RealAudio clip, you need to use great source material, high-quality equipment, and good production practices. This section surveys the steps involved in streaming a RealAudio clip.

Creating a RealAudio Clip



➤ To create a streaming RealAudio clip, follow these basic steps:

1. Capture audio source.

   You start audio production by capturing audio from a source, such as a person speaking into a microphone. You might also start with an audio source file from a compact disc, for example.

   Additional Information

   "Capturing Audio" on page 41 provides guidelines for capturing audio.

2. Optimize the audio source.

   With the audio file digitized in a common file format such as WAV or AIFF, you can use a sound editor to optimize the audio file for encoding as a streaming clip. When broadcasting live, however, you encode audio input directly from the source, optimizing the audio during capture.

   Additional Information

   See "Optimizing Audio" on page 43 for tips on editing sound. For more information on live broadcasting, read Chapter 11.

3. Encode the RealAudio clip.

   With your digitized file optimized or your live broadcast ready to go, you encode your source file in the RealAudio format. When you do this, you choose streaming bandwidths based on your target audiences.

   Additional Information

   "Encoding RealAudio with RealProducer" on page 44 provides encoding tips.

4. Deliver the RealAudio clip.

   When your presentation is ready to go, you make your RealAudio clip or broadcast available through your Web site. To combine a RealAudio clip with another streaming clip, such as a RealPix clip, you write a SMIL file.

   Additional Information

   Chapter 6 explains SMIL. See Chapter 10 for instructions on linking your Web page to a clip or a SMIL file.

Capturing Audio

A streaming clip reflects the quality of its audio source. Any quality problems within the source will affect the streaming clip as well. Because you cannot edit a broadcast, live Webcasting introduces several issues beyond those involved with delivering on-demand clips. This section will help you capture high-quality audio source files, or set up your sound equipment to deliver good broadcasts.

Source Media

If you plan to stream existing material, start with the best source possible. Use the cleanest recording with the least amount of unwanted noise. Compact discs (CDs) and digital audio tapes (DATs) are good source media, although well-recorded analog sources such as records, reel-to-reel tapes, and chrome (type II) cassettes can sound just as good. Try to avoid consumer-grade recording media such as Type I cassettes and VHS tapes.

Recording Equipment

Every piece of equipment in the audio chain—microphone, mixer, sound card, and so on—affects sound quality. If you intend to provide professional-quality audio content, invest in professional-quality audio equipment and software. Lesser equipment can add hiss and distortion, degrading sound clarity.

Shielded Cables

It is important to use high-quality, shielded cables. Using unshielded cables increases the likelihood of introducing line noise and radio frequency interference into recordings. Keep audio cables physically separated from power cords to minimize the introduction of noise. Also be sure to ground all equipment properly.

Input Levels

Setting correct input levels is crucial. All audio equipment has a signal-to-noise ratio, the ratio between the loudest possible sound the equipment can reproduce without distortion and its inherent "noise floor." Also called "clipping," distortion of this type is audible as a high-frequency crackling noise.

To get the best signal-to-noise ratio, set the input level on each audio device in the signal chain so that it uses its full range of available amplitude without distortion during the program's loudest sections. The signal chain typically includes a microphone, a mixing desk, a compressor, and a sound card. For each piece of equipment, set levels as close as possible to 0 decibels without going over that level.

Check for signal distortion at each point in the signal chain. Perform several test runs, and make sure that there are no peaks above maximum amplitude. Adjust the levels on your sound card mixer so that the input approaches but does not exceed the maximum. Be conservative, though. Levels might suddenly increase if, for instance, an interviewee suddenly speaks loudly or a crowd at a sports event roars.

Volume Levels for Live Broadcasts

When broadcasting live audio streams, it is useful to have a dynamics compressor for gain compression (not data compression). This piece of audio equipment automatically adjusts the volume level. By providing a consistent volume level, it allows you to "set and forget" the input levels to RealProducer.

Sampling Rates

Try to capture sound with a sampling width of 16 bits. RealAudio codecs have different sampling rates that produce the best sound, however. If your sound card allows it, capture audio content at the optimum sampling rate for the codec you intend to use. The RealAudio encoder will convert the file to the optimum rate if necessary, but this is recommended only for static files. For live broadcasts, use a sound card that supports the optimum rate. This avoids the overhead entailed in converting the rate while encoding sound in real time.

Additional Information

"RealAudio Codecs" on page 35 lists the optimum sampling rates for each codec.

Tip

You do not need to capture stereo sound if you plan to use a mono codec. However, many sound cards simply discard the right input channel in mono mode. If you

have a mixing desk, pan all inputs to the center so that nothing is lost during the conversion to mono.

Optimizing Audio

If you are not broadcasting audio live, you work with digitized audio source files in supported formats such as WAV or AIFF. You then edit the audio files to optimize them. To do this, you need to be familiar with the features your editing program offers. This section gives you some optimization tips you can try with your editing software.

Tip

Always keep copies of your audio source files. You cannot convert RealAudio clips back to their original source formats.

DC Offset

DC offset is low-frequency, inaudible noise that results from equipment grounding problems. If you don't remove it, it can skew the results of subsequent sound editing. Use your sound editor's **DC Offset** function immediately after recording a digital audio file.

Tip

If your editing program has this option, remove DC offset during recording. This eliminates an editing step.

Normalization

Set sensible input levels when recording, and then use normalization to maximize the levels after recording. Your streaming files sound best when your digitized source has the highest possible gain without clipping. Digital audio files that do not use their full amplitude range produce low-quality streaming clips. If the amplitude range is too low, use your sound editor to adjust the range and increase the amplitude.

Tip

Most sound editors have a **Normalize** function that maximizes levels automatically. Because some systems have trouble with files normalized to 100 percent, normalize to 95 percent of maximum, or to -0.5dB.

Dynamics Compression

Normalization maximizes the volume level of the audio file's loudest sections. Consequently, quiet sections may not encode as well. Dynamics compression evens out input levels by attenuating (turning down) the input when it rises above a specified threshold. Check your audio software for a **Compression** or **Dynamics** feature. You can control attenuation by specifying a compression ratio. This turns down the loudest sections, and you can readjust input levels accordingly.

Tip

For multipurpose dynamics compression, set the threshold to -10dB, the ratio to 4:1, and the attack and release times to 100ms. Adjust the input level to get approximately 3dB of compression and an output level of about 0dB.

Equalization

Equalization (EQ) changes the tone of the incoming signal by "boosting" (turning up) or "cutting" (turning down) certain frequencies. Using EQ, you can emphasize certain frequencies and cut others that contain noise or unwanted sound. EQ can compensate for RealAudio codecs that do not have flat frequency responses (that is, codecs for which certain frequencies are not as loud after encoding). You can therefore use EQ to make a RealAudio clip sound as close as possible to the source recording.

Tip

For voice-only content, you can make the file more intelligible by cutting frequencies below 100 Hz and carefully boosting frequencies in the 1 to 4 kHz range.

Encoding RealAudio with RealProducer

To produce RealAudio clips, you start with audio input from a live source, a digitized file, or media such as a tape or CD. You then encode a RealAudio clip from this input by using a RealAudio encoding tool. Some sound-editing programs can create RealAudio clips. But the most widely used tools for encoding RealAudio are RealProducer Basic (a free product) and RealProducer Plus, which are available from RealNetworks at:

http://www.realnetworks.com/products/producer/info.html

RealProducer on Macintosh accepts the formats widely used on that platform, whereas RealProducer on Windows and UNIX supports the formats widely used on those platforms. Check the RealProducer manual for your operating system for a list of accepted formats, which may include:

- Audio Interchange Format (.aiff)
- Audio (.au)
- MPEG-1 (.mpg)
- QuickTime (.mov)
- Sound (.snd)
- WAV (.wav)

When you encode audio clips with RealProducer, you select the target audiences you want to reach, such as 28.8 Kbps modem users. You also indicate the audio type, whether voice or music. RealProducer then determines which RealAudio codecs are best to use. RealProducer uses SureStream technology to encode the RealAudio clip so that it streams well for all your bandwidth choices, requiring only a few seconds of preroll when RealPlayer users request the clip.

Additional Information

Refer to the RealProducer user's guide or online help for step-by-step instructions for encoding RealAudio. A document explaining RealAudio codecs in-depth is at **http://service.real.com/help/library/blueprints.html**.

Note

RealAudio encoding tools other than RealProducer may not include all the features described in the following sections.

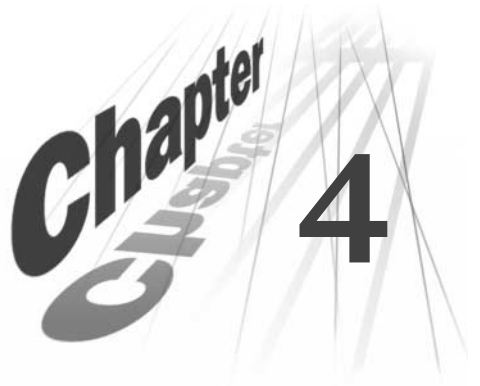
RealAudio in Multiclip Presentations

If your RealAudio clip will be played along with another clip, you may need to change RealProducer's default codec selection by adjusting the RealAudio target audience settings. Encoding mono music for 28.8 Kbps modems usually entails a 20 Kbps codec, for example, leaving no bandwidth for the second clip. RealProducer has a set of multimedia defaults, though, that lower the

clip's bandwidth. With these defaults turned on, RealProducer encodes the mono music clip at 8 Kbps, leaving 12 Kbps of bandwidth for the second clip.

Audio Compatibility with RealPlayer 5

RealProducer can include in SureStream clips a stream that RealPlayer 5 can play. This backward-compatible stream is at your lowest target bandwidth. For example, if your clip targets 28.8 Kbps modems, 56 Kbps modems, and DSL speeds, the RealPlayer 5 stream is for 28.8 Kbps modems. It uses an older RealAudio codec, and RealServer streams it to RealPlayer 5 using the older PNA protocol rather than the newer RTSP. RealPlayer 5 will receive this stream regardless of its connection speed.



PRODUCING VIDEO

RealNetworks introduced RealVideo with RealSystem 4, making streaming video available over the Internet. This chapter covers RealVideo production techniques, providing tips for capturing high-quality video, working with digitized video source files, and using RealProducer to encode your clips.

Understanding RealVideo

A video consists of two parts: the visual track and the soundtrack. In a RealVideo clip, the soundtrack is encoded with RealAudio codecs, and the visual track is encoded with a RealVideo codec. Both tracks are packaged in a RealVideo clip that, like a RealAudio clip, uses the file extension `.rm`. This section explains how RealVideo encodes a source video for streaming. This information will help you produce high-quality streaming clips.

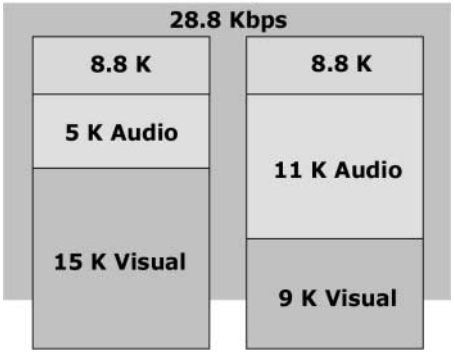
Note

Keep in mind that everything discussed about RealAudio clips in Chapter 3 also applies to the soundtracks in RealVideo clips.

RealVideo Bandwidth Characteristics

Because RealVideo uses RealAudio to encode a video's soundtrack, a chunk of the clip's bandwidth first goes toward the audio. The visual track is then squeezed into the bandwidth that's left. For 28.8 Kbps modems, for example, RealVideo clips stream at 20 Kbps, leaving 8.8 Kbps of modem bandwidth for overhead. How much bandwidth the visual track gets depends on how the audio is encoded. With a 5 Kbps RealAudio voice codec for the soundtrack, the visual track gets 15 Kbps. With an 11 Kbps music codec, though, the visual track gets just 9 Kbps.

Possible Audio and Visual Tracks in a 20 Kbps RealVideo Clip



At low bandwidths, how you encode the soundtrack can affect how the visual track looks. RealAudio music codecs typically consume more bandwidth than do voice codecs. Music's greater frequency range requires more data than does speech, so a music soundtrack consumes more bandwidth than a spoken one. A video with an audio narration might therefore look better than one accompanied by music, as there would be more bandwidth available for the visual track.

At higher streaming speeds, the soundtrack uses proportionally less of the clip's bandwidth, so differences in soundtrack encoding affect visual quality less. At speeds above 100 Kbps, you get high-quality sound that uses no more than a quarter of the clip's streaming bandwidth. The following table lists the standard RealVideo target audiences, giving the clip streaming speeds and the RealAudio codecs used for the soundtracks, broken out by audio type.

Audio Bit Rates for RealVideo Clips

		RealAudio Rate			
Target Audience	Clip Speed	Voice Only	Voice and Music	Mono Music	Stereo Music
28.8 Kbps modem	20 Kbps	6.5 Kbps	6.5 Kbps	8 Kbps	8 Kbps
56 Kbps modem	34 Kbps		8.5 Kbps		
64 Kbps single ISDN	45 Kbps	8.5 Kbps		11 Kbps	11 Kbps
112 Kbps dual ISDN	80 Kbps	16 Kbps	20 Kbps	16 Kbps	20 Kbps
Corporate LAN	150 Kbps	32 Kbps	32 Kbps	32 Kbps	32 Kbps

(Table Page 1 of 2)

Audio Bit Rates for RealVideo Clips (continued)

		RealAudio Rate			
Target Audience	Clip Speed	Voice Only	Voice and Music	Mono Music	Stereo Music
256 Kbps DSL/cable	225 Kbps	64 Kbps	44 Kbps	44 Kbps	44 Kbps
384 Kbps DSL/cable	350 Kbps		64 Kbps	64 Kbps	64 Kbps
512 Kbps DSL/cable	450 Kbps				96 Kbps

(Table Page 2 of 2)

Note

With SureStream technology, a single RealVideo clip can stream at many different speeds. For the basics of SureStream, see "SureStream RealAudio and RealVideo" on page 27.

RealVideo Frame Rates

Like RealAudio, RealVideo is "lossy," meaning that it throws out nonessential video data when encoding a clip. One way that RealVideo squeezes down clip sizes is by reducing the video's frame rate. The higher the frame rate, the smoother the motion:

- The standard frame rate for full-motion video is 24 to 30 frames per second (fps). At this speed, the human eye perceives movement as continuous—a phenomenon known as *persistence of vision*.
- A common rate for streaming video that approximates full-motion video is 15 fps. To most people, a 15 fps video flows smoothly, though not quite as fluidly as one at a higher rate.
- Below 15 fps, a video looks jerky.
- Below 7 fps, a video looks very jerky.
- Below 3 fps, a video essentially becomes a slideshow.

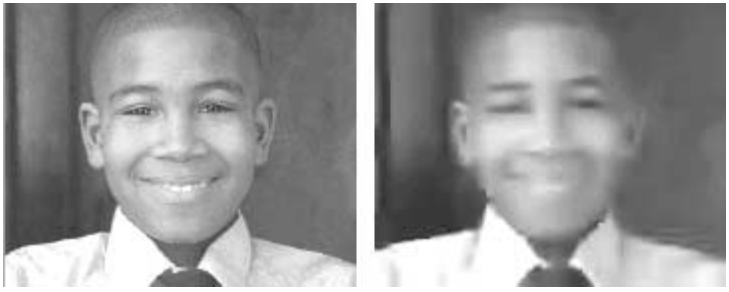
Most source videos start out at 15 to 30 fps. During encoding, RealVideo adjusts this frame rate downward as necessary, keeping the rate up in high-action scenes, reducing it in slow ones. Thus, your encoded clip will not have just one frame rate, but a mix of frame rates that varies with its content. If you follow good production practices, your clips will typically stream over slow- to medium-speed connections at 7 to 15 fps. At higher speeds, you'll get 15 to 30 fps. Many factors, though, affect a RealVideo clip's frame rate:

- The video's window size greatly affects frame rate. If you use too large of a window for your target bandwidth, you will not get a high frame rate. For more information, see "RealVideo Window Sizes" on page 51.
- RealVideo 8 provides video quality superior to that produced by older RealVideo codecs. Using an older codec may result in a lower frame rate.
- Visually complex videos that show many objects moving across the screen simultaneously are hard to encode and may result in too low a frame rate.
- RealProducer gives you an option for smoother motion. Choosing this option raises the clip's overall frame rate and reduces visual quality, whereas choosing the option for better image quality lowers the frame rate.
- In a video that has a mix of fast and slow scenes, variable bit-rate encoding (VBR) and two-pass encoding generally help the fast scenes achieve a higher frame rate. See "RealVideo Options" on page 63 for more information.
- When encoding with RealProducer Plus, you can lower the bit rate of the RealAudio codecs used for a given clip. This gives more bandwidth to the visual track, helping to raise the frame rate.
- Scalable Video Technology (SVT) enables RealPlayer to lower the frame rate as necessary during playback to lighten a computer's CPU load. For more information, see "Scalable Video Technology" on page 54.

RealVideo Clarity

In addition to changing its frame rate, RealVideo can reduce a clip's streaming size by throwing out pixel data. A video stores information about each pixel in the frame. RealVideo, on the other hand, stores data for pixel groups. When bandwidth is tight, RealVideo shoehorns pixels with slightly different RGB values into the same group. These pixels then look identical rather than nearly identical. This may result in a loss of detail if compression is too high. The following illustration compares a smooth video with one that has lost detail through too much compression.

Smooth and Distorted Video



By using good production practices as described in this chapter, you can help keep the video's clarity intact during encoding. Also note the following points:

- The video's window size affects visual clarity. If you use too large of a window for your target bandwidth, visual clarity may suffer. For more information, see "RealVideo Window Sizes" on page 51.
- When you encode with RealProducer, you can choose an option for better image quality. The video may be jerkier, though, because increasing the pixel data reduces the frame rate.
- A video with relatively stationary subjects ("talking heads") will have better visual quality than a video with rapid scene changes and a lot of movement.
- If you plan to launch a video in double- or full-screen mode as described in "Setting a Presentation's Starting Mode" on page 176, boost video clarity as much as possible during production and encoding. RealPlayer enlarges the clip by duplicating its pixels, which magnifies any defects.

RealVideo Window Sizes

The following table lists four common RealVideo clip sizes that maintain the 4:3 aspect ratio used in television. For each clip size, the table indicates the general playback quality you'll get when streaming a RealVideo 8 clip to various target audiences. "Excellent" video quality means few visual artifacts

and a frame rate that results in consistently smooth playback. Lower-quality video playback means more artifacts and a lower frame rate.

RealVideo Quality at Different Window Sizes and Bandwidths

		Video Quality for Window Size in Pixels			
Target Audience	Clip Speed	176 x 132	240 x 180	320 x 240	640 x 480
28.8 Kbps modem	20 Kbps	Excellent	Good	Fair	Poor
56 Kbps modem	34 Kbps				
64 Kbps single ISDN	45 Kbps			Good	
112 Kbps dual ISDN	80 Kbps		Excellent	Excellent	Fair
Corporate LAN	150 Kbps				Good
256 Kbps DSL/cable	225 Kbps				Excellent
384 Kbps DSL/cable	350 Kbps				
512 Kbps DSL/cable	450 Kbps				

The table shows that when streaming larger videos, you get good to excellent quality only at higher connection speeds. If you plan to stream clips over modems, you can first encode your clip at 320 by 240 pixels to test its quality. If you want better quality, shrink the video window with your editing software or with RealProducer during encoding.

Tip

When resizing a video with RealProducer, setting the RealVideo preferences to high-quality resize rather than a fast resize yields better results.

Different Window Sizes for Different Bandwidths

When you want to encode a single video for both low-speed and high-speed connections, you need to decide what video window size to use. Although with SureStream technology you can easily create a single RealVideo clip that streams at many bandwidths, using too large a window results in low frame rates for viewers with dial-up modems. Therefore, you may want to use a small video window to reach the widest audience. High-speed viewers, however, will not receive the benefit of their hefty bandwidth.

You can resolve this dilemma by creating two or more versions of your video, each for a different window size. You can use your video editing software to do this, or use RealProducer to resize or crop the video during encoding. Then encode each video for a few different streaming speeds. You can make the clips

available through separate links, or use a SMIL <switch> tag to let RealPlayer choose which version to play.

Additional Information

The basics of the <switch> tag are in "Switching Between Alternate Choices" on page 109. For a sample <switch> tag, see "Switching for Different Video Window Sizes" on page 209.

RealVideo Codecs

RealVideo 8 is the standard RealVideo codec, but you can also encode with one of two RealVideo G2 codecs. The codec you use encodes all of a clip's SureStream streams. You cannot encode half the streams with the RealVideo 8 codec, for example, and the other half with a RealVideo G2 codec.

RealVideo 8 Codec

The RealVideo 8 codec results in visual quality markedly superior to that produced by the RealVideo G2 codecs. It requires more processing power, though, so encoding a clip with it takes longer than encoding the clip with a RealVideo G2 codec. RealPlayer 8 and later can play RealVideo 8 clips. Earlier versions of RealPlayer are prompted to autoupdate to RealPlayer 8. RealNetworks recommends using this codec unless you need faster encoding performance during broadcasts, or you need to stream video to earlier versions of RealPlayer.

Note

Most users upgrade their RealPlayers soon after a new version is released, so it is generally safe to use the RealVideo 8 codec.

RealVideo G2 Codecs

The RealVideo G2 codecs are older codecs used by RealProducer G2 and RealProducer 7. These codecs encode video clips faster than the RealVideo 8 codec does, but their visual quality is poorer. For a given streaming speed, using an older RealVideo codec rather than RealVideo 8 results in either a reduced frame rate or diminished visual quality at the same frame rate. Use one of the following RealVideo G2 codecs for faster encoding during broadcasts or if you need to stream video to versions of RealPlayer that cannot upgrade to Release 8:

- RealVideo G2 with SVT

  The RealVideo G2 with SVT codec is compatible with RealPlayer G2 Update 1 (version 6.0.6) or later. RealPlayers with lower version numbers are prompted to autoupdate to the latest version of RealPlayer before viewing the clip.

- RealVideo G2

  The RealVideo G2 codec without SVT is compatible with RealPlayer G2 and later. It is not compatible with RealPlayer 5 or earlier.

RealVideo Standard Codec

In RealProducer, you cannot explicitly choose the RealVideo standard codec, which was the default codec before RealSystem G2 was released. RealProducer uses this codec solely for backward compatibility with RealPlayer 5.

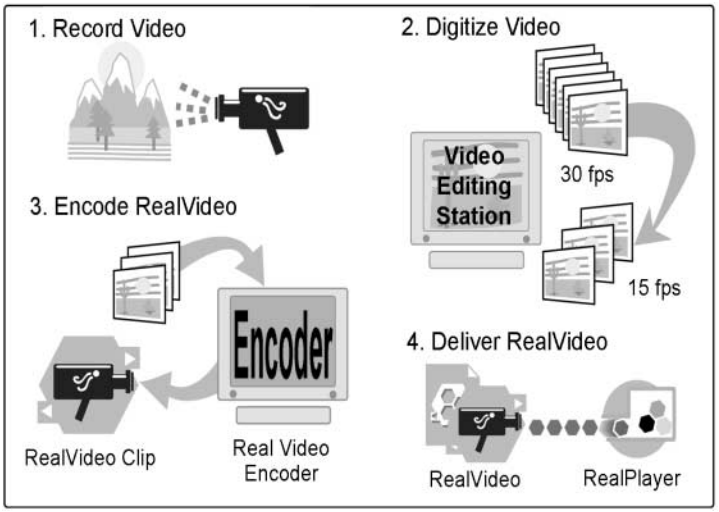
Scalable Video Technology

The RealVideo 8 and RealVideo G2 with SVT codecs include Scalable Video Technology (SVT), which scales down frame rates when clips play on slower computers. RealVideo's variable frame rate means that one scene may be encoded at 7 fps, whereas another one is encoded at 15 fps. High frame rates take a lot of processing power to decompress. Although fast PCs handle high frame rates well, slower PCs may have trouble. With SVT, RealPlayer can lower the frame rate "on the fly" to keep a PC's CPU from sputtering. So although a given scene is encoded at 15 fps, it may play on some RealPlayers at 8 fps if those computers lack the power to decompress 15 fps video clips.

Steps for Streaming RealVideo

When producing a RealVideo clip, you should choose the best source material and best equipment possible. The goal throughout the video production process is to get optimum video quality for each streaming speed.

Creating RealVideo Clips



➤ The following steps summarize how to create a RealVideo clip:

1. Capture the video content.

   To start video production, you capture the source video by shooting a scene with a video camera, for example, or gathering prerecorded content from a tape, satellite, laserdisc, or other source.

   Additional Information

   "Recording Video" on page 56 provides guidelines for shooting a video.

2. Digitize and edit the video file.

   You next digitize the video to convert it to a standard file format, such as AVI or QuickTime. With your preferred video editing software, you can then edit the video as necessary. If you are broadcasting live, however, you encode the streaming video directly from the source.

   Additional Information

   See "Digitizing Video" on page 58 for tips on video editing. For more information on live broadcasting, read Chapter 11.

3. Encode the RealVideo clip.

   With your digitized file optimized or your live broadcast ready to go, you encode your source as a RealVideo clip. When you do this, you target a network bandwidth or a set of bandwidths.

   Additional Information

   "Encoding RealVideo with RealProducer" on page 60 provides encoding tips.

4. Deliver the RealVideo clip.

   With your presentation ready to go, you make your RealVideo clip or broadcast available through your Web site. If you are combining video with another streaming clip, you write a SMIL file that assembles the pieces.

   Additional Information

   Chapter 6 explains how to create a SMIL file. See Chapter 10 for instructions on linking your Web page to a RealVideo clip or a SMIL file.

Recording Video

Read this section carefully if you intend to shoot a new video rather than use existing video content. Because video loses image quality if it's highly compressed, always start with the best video source available.

Additional Information

For pointers on recording audio, see "Capturing Audio" on page 41.

Source Media Quality

Whether you shoot a video yourself or digitize existing material, start with a high-quality video media. The following are common video formats, listed in order of descending quality:

1. Betacam SP, also known simply as Beta. This format is common among video production professionals.
2. DV, miniDV, DVCam, or DVCPro.
3. Super-VHS (S-VHS) or HI-8mm.

4. VHS, 8mm.

Video Staging

Consider the video's final frame size before you shoot the first frame. Streaming over 28.8 Kbps modems requires a small video window, so you need to frame important visual elements well. For recommended window sizes, see "RealVideo Frame Rates" on page 49.

Scene Changes and Movement

The fewer things that change from frame to frame, the sharper the image will appear in a low-bandwidth video. You can do the following to cut down on unnecessary movement:

- Use a mounted camera rather than hand-held one. This greatly reduces the movement you inadvertently introduce into the scene when recording.
- Don't have a rapidly moving object fill the entire frame. But you don't want to pull the camera back too far either. You need to find a happy medium between close-ups and panoramic shots.

Of course, you don't want to eliminate all dynamic elements. When you do include rapid movement, allow enough time for objects to resolve. Because of low frame rates and high compression, objects coming to rest may appear blurry at first. If you have a dialog box popping up on a computer screen, for example, have the box remain stationary for a few seconds so that the image resolves.

Tip

RealPix makes a great companion to RealVideo. When presenting a lecture, for example, use RealVideo to show the speaker, and use a RealPix slideshow to present visual aids such as information written on a blackboard. For more on RealPix, see "Images" on page 19.

Colors and Lighting

Bright lighting at a constant exposure keeps the foreground detail crisp. Use uniformly dark colors for backgrounds, and uniformly light colors (but not whites) for clothing. Complex textures such as paisley and stripes degrade the final image quality with unwanted visual effects.

Digitizing Video

If you are not broadcasting RealVideo streams live, you digitize the source video on your computer or video editing station. You can then edit the file with your preferred editing software before encoding it as a RealVideo clip. This section provides tips for digitizing video.

Tip

Always keep copies of the video source files. You cannot convert RealVideo clips back to their original source formats or any other streaming formats.

S-Video Output

Video playback devices commonly have at least two common output types—S-video and composite. Use S-video, as it produces better results.

Color Depth

Always use 24-bit color. Lower color resolution results in poor clips.

Digitized Video Formats

It is better to work with uncompressed formats. Otherwise, you compress the source once when you digitize it and again when you encode it as RealVideo. This double compression can decrease the image quality. Use a compressed source format only if your RealVideo encoding tool supports the file as input. You can use compressed AVI files as long as the computer used to encode RealVideo clips has the same Video for Windows driver used to compress the AVI file.

Video Capture Frame Rates

Capture the video content at 15 frames per second (fps) if your clip will stream at less than 100 Kbps. For higher-speed streaming, capture it at 30 fps. Because a RealVideo clip's frame rate varies, some scenes may be encoded at less than the capture rate depending on bandwidth constraints. SVT enables RealPlayer to lower the frame rate on computers that have low processing power, ensuring that high frame rates will not overburden slow computers.

Additional Information

For more information on SVT, see "Scalable Video Technology" on page 54.

Video Capture Screen Sizes

Unless you are short on disk space or your video capture card recommends a different window size, capture video in a window 320 pixels wide by 240 pixels high. You can capture full-motion video at the full-screen size of 640 by 480 pixels if:

- Your clips will stream at speeds higher than 100 Kbps.
- Your encoded clips will be larger than 320 by 240 pixels.
- You have a video workstation capable of digitizing full-motion, full-screen video. Standard PCs typically cannot handle this large a load.

Using your editing software or RealProducer, you can reduce the window size for your RealVideo clip if necessary. For recommended RealVideo window sizes, see "RealVideo Window Sizes" on page 51.

Computer Speed and Disk Space

Because video capture places a large burden on a computer's CPU and hard drive, be sure to use a fast computer. To avoid dropping frames during video capture, use a hard drive specially made for audio and video work. On Windows computers, you can use any video capture card that supports Video for Windows.

Use the following formula to calculate the approximate size in megabytes of a digitized video file:

$$\frac{\text{(pixel width) x (pixel height) x (color bit depth) x (fps) x (duration in seconds)}}{\text{8,000,000}}$$

Suppose you want to capture a three-minute video at 15 frames per second, with 24-bit color, in a window that is 320 by 240 pixels. As you can see from the following equation, your digitized source file would be approximately 622 MB:

```
(320) x (240) x (24) x (15) x (180) / 8,000,000 = 622 Megabytes
```

If necessary, you can conserve disk space by decreasing the window size or lowering the frame rate, or both.

Video Source File Size Limit

The Macintosh and some Windows file systems limit a single file to 2 GB (2048 MB) in size. At a 320-by-240 window size and 15 fps, this translates to about 9.5 minutes of video. Certain video production programs support the OpenDML (AVI 2.0) standard, which allows the creation of files larger than 2 GB. RealProducer may be able to accept a video source file larger than 2 GB as input, depending on the production software used to create the file.

If you plan to produce long videos or videos with a large window size, check whether or not your video production software is limited to a 2 GB output file size. If it is not limited to 2 GB, create a video file larger than 2 GB and test to determine if RealProducer can accept the file as input.

Additional Information

Search RealNetworks' Knowledge Base for "AVI limit" at **http://service.real.com/kb/index.html**.

Tip

If you are limited to 2 GB for the video source file and you need to produce a larger video, you can create separate video source files (each 2 GB or smaller) and encode them as separate RealVideo clips. Then, merge the clips using RealProducer's editing tools. See the *RealProducer User's Guide* for more information.

Encoding RealVideo with RealProducer

To produce RealVideo clips, you start with video input from a live source, a digitized file, or media such as a tape or CD. You then encode a RealVideo clip from this input using a RealVideo encoding tool. The most widely used tools for encoding RealVideo are RealProducer Basic (a free product) and RealProducer Plus, which are available from RealNetworks at:

http://www.realnetworks.com/products/producer/info.html

RealProducer on Macintosh accepts the input formats widely used on that platform, whereas RealProducer on Windows or UNIX supports the formats widely used on those platforms. Check the RealProducer manual for your operating system for a list of accepted formats, which may include:

- AVI (.avi), uncompressed (recommended) or compressed

- MPEG-1 (.mpg)
- QuickTime (.mov), uncompressed (recommended) or compressed

When you encode RealVideo clips, you choose a bandwidth target or set of targets, then set parameters such as audio type (voice or music) and an emphasis on either smooth motion or sharp images. RealProducer then decides which RealAudio codecs are best to use for the soundtrack. It uses SureStream technology to encode the RealVideo clip for all your bandwidth choices, ensuring that each stream requires only a few seconds of preroll.

Additional Information

Refer to the RealProducer manual or online help for step-by-step instructions for encoding clips. A document explaining RealVideo codecs and options in-depth is at **http://service.real.com/help/library/blueprints.html**.

Note

RealVideo encoding tools other than RealProducer may not include all the features described in the following sections.

RealVideo Streaming Speeds

For all practical purposes, RealVideo's streaming bit rate is infinitely scalable. You can encode RealVideo at any speed you want, from 20 Kbps to hundreds or thousands of kilobits per second. Plus you can encode precisely at any bandwidth you choose, such as 89 Kbps, 117 Kbps, 575 Kbps, 1.5 Mbps, and so on. With RealProducer Plus, you do this by adjusting RealVideo's target audience settings.

If your RealVideo clip will play in parallel with another clip, you may need to lower RealVideo's bit rate for each target audience. This way, the clip does not consume all of a connection's bandwidth. For a 56 Kbps modem, for example, RealProducer standardly encodes the clip to stream at 34 Kbps. Using the RealVideo target audience settings, you can lower this value to 20 Kbps, for example, leaving 14 Kbps free for another clip.

Whenever you lower RealVideo speed, the new bit rate includes the rate of the RealAudio codec used for the soundtrack. If you use a 32 Kbps RealAudio codec and set a total RealVideo bit rate of 36 Kbps, you will not have much bandwidth left for the visuals (maybe a frame every few seconds). You'll need to select a lower-bandwidth RealAudio codec.

There are no strict rules for the ratio of soundtrack bandwidth to total RealVideo bandwidth. The ratio depends entirely on whether you want to emphasize the audio track or the visuals. RealVideo normally allots no more than 25 percent of the total bandwidth to the soundtrack because the visual information contains much more data. At bandwidths of several hundred kilobits per second, you can use considerably less bandwidth for the soundtrack in relation to the visual track.

Video Compatibility with RealPlayer 5

RealProducer can include in SureStream clips a stream that RealPlayer 5 can play. This backward-compatible stream is set for your lowest target bandwidth. For example, if your clip targets 28.8 Kbps modems, 56 Kbps modems, and DSL speeds, the RealPlayer 5 stream is for 28.8 Kbps modems. It uses older RealAudio and RealVideo codecs, and RealServer streams it to RealPlayer 5 using the older PNA protocol rather than the newer RTSP.

RealVideo Filters

RealProducer's preferences include filtering options you can use when encoding RealVideo clips. These filters remove artifacts that appear in the encoded clips because of the methods used to create the source video.

Noise Filters

Noise distortion is similar to the "snow" that often shows up on TV screens as a result of signals received over an antenna. RealProducer has low and high noise filters you can use if your source video clip has a little noise or a lot of noise, respectively. The best way to eliminate noise, though, is to follow good production practices, as described in "Recording Video" on page 56.

Resizing Filters

When you resize a RealVideo clip with RealProducer, you can use a resizing filter that performs either a fast resize or a high-quality resize. These resizing options affect the source video only when you make the clip smaller. They tell RealProducer to throw out video data by using either a quick method (fast resize) or a complex analysis (high-quality resize). A high-quality resize results in a superior image, but it also lengthens the encoding time.

Inverse-Telecine Filter

The Inverse-Telecine filter is for cinematic film that was transferred to NTSC video. It has no effect on PAL video. Film is usually photographed at 24 frames per second (fps), and the film-to-video conversion (called "telecine") duplicates some frames to achieve the NTSC standard of 30 fps. To strip out redundant frames, use the inverse-telecine filter when encoding NTSC video that was transferred from film and has a frame rate of 25 to 30 fps.

De-Interlace Filter

The de-interlace filter removes jaggedness from interlaced video. It is useful only for interlaced source video that is at least 240 lines high. Typical source video used for television is 480 lines high. If you digitize the video with a video capture card that captures at 240 lines or fewer, the card throws out either the odd or the even lines, de-interlacing the video itself.

RealVideo Options

The RealProducer preferences include several RealVideo encoding options that affect the quality of RealVideo clips by modifying RealProducer's encoding methods.

Variable Bit Rate Encoding

Variable bit rate (VBR) encoding is optional for RealVideo. It varies a clip's *playback* bit rate while keeping the *streaming* bit rate constant. This gives more bandwidth to scenes that are hard to compress and less to scenes that are easy to compress. In a video that streams at 225 Kbps, for example, one second may get 150 Kb of data while another gets 300 Kb. VBR encoding generally provides video quality superior to that achieved with constant bit rate (CBR) encoding, making the most difference in videos with a mix of high- and low-action scenes.

Two-Pass Encoding

With two-pass encoding, RealProducer runs through the source video once to gather information. It then makes a second pass to encode the RealVideo streams. Two-pass encoding helps the most with variable bit rate (VBR) encoding. With this type of encoding, RealProducer analyzes the entire source video first to determine how best to vary the playback bit rate. Without it, RealProducer analyzes small sections of the source video during encoding, creating a string of VBR sections within the clip.

Loss Protection

RealProducer's loss-protection feature adds error-correction data to RealVideo streams, helping them maintain quality in lossy environments. RealProducer adds as much error-correction data as it can without lessening the video quality. Although you'll get more benefit from this feature when streaming across the Internet rather than over an intranet, it is useful to turn loss protection on for all encoded content.

PRODUCING ANIMATION

Macromedia Flash makes it easy to put animation on the World Wide Web. Streamed through RealSystem, Flash clips can deliver visually arresting animations and e-commerce applications to RealPlayer. This chapter provides guidelines for creating and optimizing Flash clips that stream to RealPlayer. For instructions on developing Flash animation, refer to the Flash user's guide.

Additional Information

Learn more about Flash from Macromedia's Web site at **http://www.macromedia.com/software/flash**.

Understanding Flash

Flash is well-suited for linear presentations that have a continuous audio track and animated images synchronized along a timeline. Such presentations could include:

- Demonstrations, training courses, and product overviews.
- Full-length cartoons for entertainment and education.
- Product advertisements.
- Movie trailers.
- Karaoke.

With Flash commands, you can build interactive icons and forms for:

- Electronic commerce.
- On-screen navigation.
- Internet radio tuners.
- E-mail registration.

This section explains how Flash works with RealSystem. This knowledge will help you produce high-quality streaming animation.

RealSystem Software Versions for Flash

Streaming Flash version 3 or 4 to RealPlayer requires RealServer 8 or later. Earlier versions of RealServer stream only Flash 2. RealPlayer 8 or later is required to play Flash 3 or 4 clips. RealPlayer G2 or 7 will autoupdate to the latest RealPlayer release when it encounters a Flash 3 or 4 clip.

RealPlayer does not support the Flash 5 Player format. You can develop your animation with the Flash 5 program, but your exported Flash Player clip must be in the Flash 2, 3, or 4 format. Note that the Flash 5 program can automatically export and tune your clips in the Flash 4 format for streaming to RealPlayer.

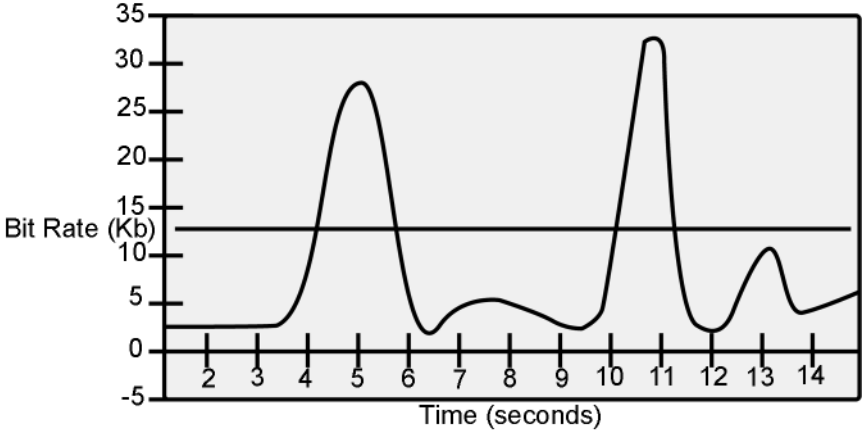
Flash Bandwidth Characteristics

As with any streaming clip, you develop a Flash clip with a target audience bandwidth in mind. The table "Maximum Streaming Rates" on page 24 lists the highest rate at which your Flash clip should stream for various network connection speeds. Keep in mind, too, that if your Flash clip streams along with other clips, the combined streaming speed of all the clips should not exceed the maximum speed for the target audience. This helps ensure that your presentation does not rebuffer frequently.

Because most Internet users have 28.8 or 56 Kbps modems, RealNetworks recommends that you target dial-up modem audiences. Fortunately, Flash clips streamed over a 28.8 Kbps modems can have a visual impact comparable to that of a video streaming at a significantly higher bit rate. This is because Flash clips transmit vector information rendered by the viewers' computers. Hence, the quality of Flash animation depends more on a computer's CPU and graphics capabilities than on the amount of streamed data.

Because it is vector-based, Flash does not consume bandwidth evenly. When a scene starts, for example, its groups and symbols are streamed, requiring a lot of data transfer. After that, only lightweight instructions for manipulating groups and symbols are needed. This following illustration shows a Flash clip that targets a streaming speed of 12 Kbps. At 5 and 10 seconds into the clip's timeline, bandwidth use spikes because the clip needs more than 12 kilobits of data to change scenes or to introduce new objects in a key frame.

Sample Bit Rate Requirement for a Streaming Flash Clip



If it encounters spikes, RealPlayer buffers the data, delaying playback until all of the necessary data has arrived. For your clip to stream well, you must eliminate spikes by tuning the finished clip. Tuning the clip also sets the clip's streaming bit rate and preroll. The Flash 5 program can export and tune a clip automatically. Or, you can tune an exported clip manually with the Flash tuner included in the RealSystem Authoring Kit, which you can download from the following Web site:

http://www.realnetworks.com/products/authkit/index.html

Tip

You will not know how well your clip streams until you tune it. Because you may need to revise the animation to make the clip stream well for your chosen audience, export and tune the animation frequently as you develop it.

Flash Clip Size

Tuning your Flash clip guarantees that it streams at your chosen bit rate. If your animation is too complex, however, tuning it to a low bit rate may cause an unacceptably high preroll in RealPlayer. The best way to guarantee a low preroll is to keep the ratio of clip size to clip length low. The following are tips for keeping the Flash clip size as small as possible as you develop your animation:

- Reduce key frames.

  Excessive key frame changes increase bandwidth consumption. Minimize the number of key frames and simplify the objects within key frames.

- Use symbols instead of groups.

  Flash stores a symbol once and can then refer to it at any time, with each reference adding little to the file size. However, it stores a group definition each time the group is used. Using a group three times, for example, stores the same data in the file three times. Using symbols instead of groups can therefore reduce file size significantly.

- Simplify elements.

  Simplify the elements that you draw or import into Flash. Under **Modify>Curves**, use the **Smooth** and **Straighten** commands on lines and curves to strip away unneeded point and path information. This reduces the amount of data stored for each element. Use **Optimize** to optimize the data reduction while maintaining acceptable screen appearance. Because screen resolution is lower than print resolution, you can eliminate minute details without compromising appearance.

- Do not include event sound.

  As described in "Adding Audio to Flash" on page 69, RealPlayer does not support event sounds. Including event sounds in the exported Flash Player clip wastes bandwidth. Either remove event sounds from the source Flash file (`.fla`) or export the Flash Player clip (`.swf`) without sound.

- Adjust JPEG quality when exporting.

  If your animation has imported graphics, set the JPEG quality to no greater than 50—possibly as low as 30—when exporting the `.fla` file to a `.swf` clip.

Flash CPU Use

Bandwidth use is not the only consideration when developing Flash animation. Because it is vector-based, Flash performs complex calculations on the user's computer to display the animation. Operations that require many calculations on top of the computer's normal load may adversely affect playback. Newer computers typically have processors that are fast enough to handle Flash and other clips streamed in parallel, but older computers may

not have this capacity. To support the widest audience possible, follow these recommendations to reduce Flash CPU requirements:

- Reduce the frame rate.

  Macromedia recommends a Flash frame rate of 12 frames per second (fps). If you combine a Flash clip with another clip that needs considerable processing power, though, you may need to lower this frame rate to accommodate slow computers. Try 9 fps or 7 fps when combining Flash with RealAudio, for example. These rates provide acceptably smooth motion without overburdening most processors.

- Optimize tweening.

  The tweening process interpolates the motion between key frames. Interpolating multiple objects and color effects at the same time will adversely affect playback. Other actions related to tweening that slow down playback include changing large areas of the screen between frames and using gradient fills.

- Decrease the number and size of objects moving simultaneously.

  RealPlayer must redraw areas where action occurs, thus consuming CPU power. To minimize this, localize tweening to a small portion of the screen so that the entire screen does not have to be redrawn. This way, file size remains the same, but only one part of the screen is redrawn.

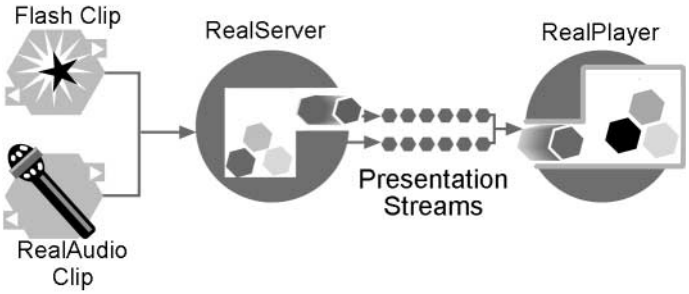
Adding Audio to Flash

RealPlayer does not support audio embedded in a Flash clip, so it will not play event sounds for mouseovers and button clicks, for example. You can include sound for a Flash clip only by exporting the soundtrack and encoding it as a RealAudio clip that plays in parallel with Flash. Therefore, audio tracks are beneficial primarily for linear clips such as a cartoons, rather than for interactive applications.

Tip

Flash provides different methods for incorporating sound into an animation. Use the stream synchronization setting.

A Flash Soundtrack Uses a Separate RealAudio Clip



To create separate animation and audio clips, you first synchronize your animation with an imported sound file, such as a WAV or AIFF file. You then export two files:

- A Flash Player clip that contains no sound.
- A soundtrack that you encode as a RealAudio clip.

Using SMIL, you synchronize both clips for streaming. The Flash 5 program can create a SMIL file and export your soundtrack as a RealAudio clip automatically. If you use an earlier version of Flash, you need to export the soundtrack manually, encode it as a RealAudio clip, and write the SMIL file. "Streaming a Flash Clip" on page 78 summarizes this exporting process.

Bandwidth Division Between Flash and RealAudio

When you export and encode your Flash soundtrack as a SureStream RealAudio clip, all viewers get the same Flash clip, but they get different RealAudio streams depending on their network connection speeds. For any network connection, determining your Flash and RealAudio clip speeds is a two-step process:

1. Decide which RealAudio codecs to use to encode the soundtrack. All codecs are listed in "RealAudio Codecs" on page 35.
2. For your lowest-speed target audience, subtract the lowest RealAudio streaming speed from the target's maximum streaming speed to get the Flash clip's maximum streaming speed.

Additional Information

The table "Maximum Streaming Rates" on page 24 lists the streaming speeds for various network connection speeds.

Targeting 28.8 Kbps Modems

The following table lists possible RealAudio and Flash bit-rate combinations for 28.8 Kbps modems, which have a maximum streaming speed of 20 Kbps. If you choose an 8 Kbps music codec for RealAudio, for example, you have 12 Kbps of streaming bandwidth left for Flash.

Bandwidth Divisions between RealAudio and Flash at 20 Kbps

Soundtrack Type	RealAudio Codec	Flash Maximum Speed
Voice	5 Kbps Voice	15 Kbps
	6.5 Kbps Voice	13.5 Kbps
	8.5 Kbps Voice	11.5 Kbps
Music	6 Kbps Music	14 Kbps
	8 Kbps Music	12 Kbps
	11 Kbps Music	9 Kbps

Targeting 56 Kbps Modems

Suppose you want to reach 56 Kbps modems, which have a maximum streaming speed of 34 Kbps. The following table lists some RealAudio codecs you can use, indicating for each codec the streaming speed left for the Flash clip.

Bandwidth Divisions between RealAudio and Flash at 34 Kbps

Soundtrack Type	RealAudio Codec	Flash Maximum Speed
Voice	6.5 Kbps Voice	27.5 Kbps
	8.5 Kbps Voice	25.5 Kbps
	16 Kbps Voice	18 Kbps
Music	11 Kbps Music	23 Kbps
	16 Kbps Music	18 Kbps
	20 Kbps Music	14 Kbps

Targeting Both 28.8 and 56 Kbps Modems

To target both 28.8 and 56 Kbps modems, decide first how to reach the 28.8 Kbps audience. For a voice soundtrack, for example, you might use a 6.5 Kbps RealAudio voice codec, leaving 13.5 Kbps for Flash. To reach 56 Kbps modems, you would encode the soundtrack as a SureStream RealAudio clip using both the 6.5 Kbps voice codec and a 16 Kbps voice codec. Users with 56 Kbps modems then get 16 Kbps of RealAudio data along with the 13.5 Kbps Flash clip. This puts the streaming speed for this combination at 29.5 Kbps, a little less than the 34 Kbps maximum.

Tips for Choosing RealAudio Codecs

Here are some tips for selecting a RealAudio codec to use with a streaming Flash clip:

- If sound quality takes precedence, use the fastest RealAudio codec that still leaves enough bandwidth for acceptable animation.
- When animation is complex, use low-speed RealAudio codecs targeted for voice. This increases the bandwidth available for the animation.
- If possible, do not select the lowest-speed RealAudio codec. SureStream clips include a duress stream that is used if the connection bandwidth falls. An 8 Kbps music clip, for example, includes a 6 Kbps duress stream. If you encode the clip using just the 6 Kbps codec, RealPlayer will have no duress stream to fall back on.
- To encode a RealAudio clip with exactly the codec you want, you may need to change the RealAudio default target audience settings. You need RealProducer Plus to do this, because RealProducer Basic does not allow changes to the default settings.

Using Interactive Flash Commands

Because RealPlayer supports all Flash 3 and Flash 4 commands, you can make your presentations interactive by adding buttons and forms. In some cases, Flash commands work differently in RealPlayer than in the Flash browser plug-in. This section provides guidelines for using Flash commands with RealPlayer presentations.

Flash Clip Timeline Commands

Flash has several commands you can use to control the Flash clip's timeline. In a Flash 3 or 4 clip, these commands affect only the Flash clip. The presentation and all other clips playing along with the Flash clip continue through their timelines normally. In a Flash 2 clip, these commands affect all clips playing in RealPlayer.

Interactive Flash Commands

Command	Action
`Play`	Flash clip playback begins or resumes.
`Stop`	Flash clip pauses until a `Play` command is issued. With a Flash 3 or 4 clip, all other clips play normally. With a Flash 2 clip, all other clips pause.
`Go To and Stop`	RealPlayer seeks to the designated frame in the Flash clip and pauses. The Flash clip timeline resumes on a `Play` command. With a Flash 3 or 4 clip, all other clips play normally. With a Flash 2 clip, all other clips seek to the same point in the presentation timeline and then pause. See also "Go To Commands" on page 74.
`Go To and Play`	RealPlayer seeks to the designated frame in the Flash clip, buffers the clip preroll, and begins playback. With a Flash 3 or 4 clip, all other clips play normally. With a Flash 2 clip, all other clips seek to the same point in the presentation timeline and then resume playback.
`Get URL`	RealPlayer sends the URL to the browser. Because the user has to return to the animation manually, use this only at the end of a clip. Also note that a SMIL file can define clickable hyperlinks that overlay a Flash clip. See "Linking to Other Media" on page 112.

RealPlayer Commands

As noted in the preceding table, commands such as **Play**, **Stop**, and **Go To** in Flash 3 and Flash 4 clips affect only the Flash clip. Using Flash's **Get URL** command, though, you can play, stop, or pause all clips playing in RealPlayer. You can also launch a URL in a new RealPlayer window. You do this by sending RealPlayer a command (rather than a URL) through **Get URL**.

Seeking Into a Presentation

The following value for **Get URL** instructs RealPlayer to seek to the specified time in the presentation timeline:

```
command:seek(time)
```

For example, the following command instructs RealPlayer to seek to 1:35.4 in the presentation timeline:

```
command:seek(1:35.4)
```

The time format is as follows:

```
dd:hh:mm:ss.xyz
```

Here, `dd` is days, `hh` is hours, `mm` is minutes, `ss` is seconds, `x` is tenths of seconds, `y` is hundredths of seconds, and `z` is milliseconds. Only the `ss` field is required. When the time value does not include a decimal point, RealPlayer reads the last field as the seconds. For example, `1:30` means 1 minute and 30 seconds, whereas `1:30:00` means 1 hour and 30 minutes. Note that all of the following commands are equivalent. Each seeks to the point 90 minutes into the presentation timeline:

```
command:seek(1:30:00.0)
command:seek(90:00)
command:seek(5400)
```

Playing, Pausing, or Stopping a Presentation

The following values for **Get URL** cause RealPlayer to play, pause, or stop the presentation, respectively:

```
command:play()
command:pause()
command:stop()
```

Popping Up New RealPlayer Windows

Using the **Get URL** command, you can open streaming presentations in new RealPlayer windows. In RealPlayer 7 or later, you can open as many player windows as the computer's CPU and memory allow. For complete information, see "Popping Up New RealPlayer Windows" on page 127.

Go To Commands

Use **Go To** commands only when adding interactivity to a Flash clip. Do not use them to advance from one scene to the next. When you export your animation in the Flash Player format, scenes are concatenated so that the animation flows from one scene to the next. A **Go To** command causes RealPlayer to seek to the target frame. If RealServer has not yet streamed the target frame, RealPlayer halts clip playback, issues a seek request to RealServer, and rebuffers the new data as it comes in.

When you use at least one **Go To** command in a Flash 3 or 4 clip, RealPlayer caches the entire clip in memory. It assumes that the clip is interactive and that the **Go To** commands are meant to move the viewer from one part of the clip to another based on input such as a button click. After RealServer has streamed the frames containing the **Go To** command and its target, RealPlayer does not need to rebuffer the clip when the viewer gives the command.

Using a **Go To** command raises RealPlayer's memory requirement for playing the clip. This is generally not a problem, because Flash memory requirements are low. When authoring long, linear animations, though, avoid using **Go To** commands whenever possible. When no **Go To** commands are present, RealPlayer discards clip data it no longer needs. This helps guarantee good-quality playback on computers that are low on available memory.

Load Movie Commands

RealPlayer imposes a restriction on using Flash's **Load Movie** command to import a second Flash clip into a clip that is playing. If the clips use the RTSP protocol, RealServer stops the first clip and streams the second clip as a new RealPlayer presentation rather than streaming the second clip as part of the initial presentation. The **Load Movie** command works properly only when clips are downloaded with HTTP. There are two ways to manage this:

- Stream the first clip with RTSP by using `rtsp://` in the SMIL or Ram file URL for the clip. In a **Load Movie** command, use a fully qualified HTTP URL for the clip. RealPlayer will then request the clip with the given URL. This is the preferred solution because the first clip uses RTSP, which is a better protocol for streaming.

  Tip
  RealServer supports both RTSP and HTTP. You can therefore put all clips in the same RealServer directory, streaming the first one with RTSP and all of the others with HTTP. Just be sure not to include `/ramgen/` in the URLs used with the **Load Movie** command.

- The second solution is to download all clips by using HTTP. Use `http://` in the SMIL or Ram file URL to the initial clip. In a **Load Movie** command, you can then refer to an imported clip using just its file name. RealPlayer requests subsequent clips using the same HTTP URL (except for the different file names) used to download the first clip.

Tip

If your presentation does not use SMIL, use a Ram file instead of Ramgen to list the HTTP URL to the first clip. RealServer's Ramgen utility adds `/ramgen/` to the first clip's URL. When RealPlayer reuses this URL, the `/ramgen/` component starts a new presentation.

Timeline Slider Activity with Multiple Clips

If your presentation includes multiple Flash clips integrated with **Load Movie** commands, the RealPlayer slider reflects only the first clip's timeline. Suppose that a clip plays for five minutes and then loads another clip. The RealPlayer slider is active only for the five minutes the first clip plays. After that, the second clip plays normally, but RealPlayer indicates that the presentation has finished by resetting the timeline slider and disabling the stop button. Viewers can still perform interactive functions and stop the second clip by using Flash's contextual menu, though.

Using SMIL Instead of Load Movie

You need to use the **Load Movie** command to insert a new Flash clip into a Flash clip that is already playing. You do not need to use this command to play two or more Flash clips in sequence, though. Instead, you can use SMIL to define the sequence. This overcomes the URL and timeline limitations described above. To play two clips in sequence, for example, you write a SMIL file that looks like the following:

```
<smil>
 <body>
  <seq>
   <animation src="rtsp://realserver.example.com:554/media/cartoon1.swf"/>
   <animation src="rtsp://realserver.example.com:554/media/cartoon2.swf"/>
  </seq>
 </body>
</smil>
```

You can also use SMIL to combine each Flash clip with a RealAudio clip. The example below has two clip groups that play in sequence. Each clip group is composed of a Flash clip and a RealAudio clip played in parallel:

```
<smil>
 <body>
  <seq>
   <par>
```

```
      <animation src="rtsp://realserver.example.com:554/media/cartoon1.swf"/>
      <audio src="rtsp://realserver.example.com:554/media/sound1.rm"/>
    </par>
    <par>
      <animation src="rtsp://realserver.example.com:554/media/cartoon2.swf"/>
      <audio src="rtsp://realserver.example.com:554/media/sound2.rm"/>
    </par>
  </seq>
 </body>
</smil>
```

Additional Information

For information on SMIL, see Chapter 6.

Secure Transactions

With Flash forms, you can build transaction functionality directly into Flash clips streamed to RealPlayer. This lets you add e-commerce capability to your presentation, for example. If the Flash clip connects to a secure server, RealPlayer transmits the encrypted information through the viewer's browser. RealPlayer does not support encryption directly, so the viewer needs to have a browser that supports it. Any encrypted response sent back by the secure server displays in the browser.

Note

Because RealPlayer does not support secure transactions, you should not send an HTTP **POST** or **GET** command to a secure server if you intend for the server's response to come back to RealPlayer. For example, do not connect to a secure server by using Flash's **Load Variables** or **Load Movie** command.

Mouse Events

RealPlayer tracks certain mouse events differently than does the Flash plug-in used with browsers. Although this does not change how you build a streaming Flash presentation and it will not affect most viewers, you should be aware of this behavior.

The Flash browser plug-in records mouse events that occur outside of the Flash area. For example, a user may click and hold on an icon, drag the pointer out of the Flash area, and release the mouse button. In this case, the Flash

browser plug-in knows that the mouse button has been released. RealPlayer, however, does not record mouse events that occur outside of its Flash region. Instead, it assumes that the button is still held down when the pointer returns to the Flash region.

Streaming a Flash Clip

This section summarizes the process for streaming a Flash clip. The Flash 5 program can export a RealAudio clip, a tuned Flash Player clip, and a SMIL file automatically. If you use Flash 5, refer to your Flash user's guide for instructions on exporting and tuning clips. Macromedia's Web site also provides information about exporting clips with Flash 5:

http://www.macromedia.com/support/flash/publishexport/realplayer/realplayer.html

If you are using a version of the Flash program prior to version 5, you must perform the following export and tuning steps manually.

➤ To create a streaming Flash clip manually:

1. Export the Flash Player clip.

   RealServer streams only the Flash Player format (`.swf`), which is a compressed version of the animation. You cannot stream the Flash source file format (`.fla`). If your animation includes a soundtrack, disable the audio stream when you export the clip. Refer to the Flash user manual for step-by-step instructions on the exporting a Flash Player clip.

   Note
   Keep in mind that RealPlayer plays the Flash 4, 3, and 2 Player formats. It does not play clips in the Flash 5 format.

2. Tune the Flash Player clip.

   With the Flash tuner, set the clip's streaming bit rate. This necessary step also eliminates bandwidth spikes that can cause rebuffering. The Flash tuner is included in the RealSystem Authoring Kit, available at this Web site:

   http://www.realnetworks.com/products/authkit/index.html

3. Export the soundtrack.

   If your animation includes a soundtrack, export the soundtrack as a Windows WAV file or Macintosh QuickTime file. If exporting to QuickTime (or any other video format), set low height and width attributes to minimize disk space use.

4. Encode the soundtrack as RealAudio.

   Encode the exportedWAV or QuickTime soundtrack in the RealAudio format with RealProducer, using the file extension `.rm`.

5. Deliver the Flash presentation.

   Transfer your clips to RealServer. Then write the SMIL and Ram files necessary to stream the presentation.

   - Streaming a single Flash clip

     If your have a single Flash clip, your RealServer administrator can give you the URL to use in your Web page's hyperlink to the clip. If the RealServer does not use Ramgen, or you are delivering the clip through a Web server, you need to write a Ram file.

     Additional Information

     For more on RealServer and Ramgen, see "Streaming Clips from RealServer" on page 166. Ram files are described in "Creating a Ram File Manually" on page 173.

   - Streaming a Flash clip with another clip

     If your presentation has multiple clips, you write a SMIL file that organizes the presentation and gives the clip URLs. You next link your Web page to the SMIL file. In its simplest form, the SMIL file gives the full URLs to the clips and specifies that the clips play in parallel. The following example is for a Flash clip that plays in parallel with a RealAudio soundtrack:

```
<smil>
  <body>
    <par>
      <audio src="rtsp://realserver.example.com:554/media/sound.rm"/>
      <animation src="rtsp://realserver.example.com:554/media/cartoon.swf"/>
    </par>
  </body>
</smil>
```

You can also use SMIL to define hypertext links, create timing offsets between clips, or add presentation information such as title, author, copyright, and abstract. For complete information on SMIL, see Chapter 6.

ASSEMBLING A PRESENTATION WITH SMIL

When your streaming presentation contains multiple clips—such as a slideshow and a video played together—you use Synchronized Multimedia Integration Language (SMIL) to coordinate the parts. Pronounced "smile," SMIL is a simple but powerful markup language for specifying how and when clips play. After writing the SMIL file, you put it on RealServer and link your Web page to it as described in Chapter 10.

Additional Information

Once you are familiar with SMIL, refer to "Appendix D: SMIL Quick Reference" beginning on page 219.

Understanding SMIL

With your clips in their streaming formats, you put your presentation together with SMIL. A SMIL file is not required to stream just one clip. But when you have multiple clips, SMIL's simple markup language specifies how and when the clips play. There are many advantages to using SMIL:

- Avoid using container formats.

  Because RealSystem can stream many media formats, you do not need to merge clips into a single streaming file. To alter your presentation, for example, you simply edit the SMIL file rather than merge the clips again into a different container file.

- Use clips in different locations.

  Because a SMIL file lists a separate URL for each clip, you can put together presentations using clips stored on any server.You can use a video clip on a RealServer, for example, and a text clip on a Web server.

- Time and control a presentation.

  The SMIL file lets you easily control the presentation timeline. You can start an audio clip playing at 2.5 seconds into its internal timeline, for example, without changing the encoded clip.

- Lay out a presentation.

  When your presentation includes multiple clips, such as a RealVideo clip playing simultaneously with a RealPix slideshow, you use SMIL to define the layout.

- Stream clips in multiple languages.

  A SMIL file can list different language options for clips. To create a video with soundtracks in different languages, for example, you produce one video clip with no soundtrack, then create a separate audio clips for each language. Your Web page needs just one link to the SMIL file. When a visitor clicks that link, the visitor's RealPlayer chooses a soundtrack based on its language preference.

- Reach viewers at multiple bandwidths.

  A SMIL file can also list presentation choices for different bandwidths. RealPlayer then chooses which clips to receive, based on its available bandwidth. You can thereby support multiple connection speeds through a single hypertext link, rather than separate links for modem users, ISDN users, T1 users, and so on.

  Note

  RealSystem's SureStream technology also lets you support multiple bandwidth connections within a single clip. For more information, see "SureStream RealAudio and RealVideo" on page 27.

- Put together customized presentations.

  Because a SMIL file is a simple text file, you can generate it automatically for each visitor. You can therefore create different presentation parts, assembling a customized SMIL file for each visitor, based on preferences recorded in the visitor's browser.

- Include ads in presentations.

  You can use SMIL along with RealServer's advertising extension to insert ads into your presentation. RealServer can deliver image banner ads, as

well as media ads in formats such as RealVideo and Flash. Chapter 9 explains how to use SMIL to insert ads in streaming presentations.

Advantages of Using SMIL

Creating a SMIL File

You can create a SMIL file (extension .smil) by using any text editor or word processor that can save output as plain text. If you are familiar with HTML markup, you will pick up SMIL quickly. In its simplest form, a SMIL file lists multiple clips played in sequence:

```
<smil>
  <body>
    <audio src="rtsp://realserver.example.com/one.rm"/>
    <audio src="rtsp://realserver.example.com/two.rm"/>
    <audio src="rtsp://realserver.example.com/three.rm"/>
  </body>
</smil>
```

General SMIL Rules

SMIL has many similarities to HTML, but also some important differences. When you create a SMIL file, keep the following general rules in mind.

Markup Starts with <smil> and Ends with </smil>

The SMIL markup must start with a <smil> tag and end with the </smil> closing tag. All other markup appears between these two tags:

```
<smil>
 ...all other SMIL markup...
</smil>
```

Body Section Is Required but Header Section Is Optional

A SMIL file can include an optional header section defined by <head> and </head> tags. It requires a body section defined by <body> and </body> tags:

```
<smil>
  <head>
    ...optional section with all header markup...
  </head>
  <body>
    ...required section with all body markup...
  </body>
</smil>
```

The header section is used to specify presentation information, as explained in "Adding Presentation Information" on page 118, and to define clip layout, as described in "Laying Out Multiple Clips" on page 99.

Lowercase Text is Required for Tags and Attributes

SMIL tags and attributes must be lowercase.

Some Tags Must Close with a Forward Slash

A tag that does not have a corresponding end tag (for example, the <smil> tag has the end tag </smil>) must close with a forward slash. For example:

```
<audio src="first.rm"/>
```

Attribute Values Are Enclosed in Double Quotation Marks

Attribute values, such as "first.rm" in the preceding paragraph, must be enclosed in double quotation marks. Each file name in SMIL must reflect the file name exactly as it appears on the server. File names can use uppercase, lowercase, or mixed case.

SMIL Files Use the Extension .smil or .smi

Save your SMIL file with the extension .smil or .smi. The .smil extension is preferable, as it reduces the potential for conflict with other files. Do not include spaces in the file name. For example, you can have the file my_presentation.smil but not the file my presentation.smil.

Coded Characters Are Used in Headers

You need to use codes to add quotation marks, apostrophes, ampersands, or angle brackets to content in a SMIL header, such as a presentation title. See "Using Coded Characters" on page 121.

HTML-Style Comments Are Allowed

As in HTML, the SMIL comment tag starts with **<!--** and ends with **-->**. The ending does not include a forward slash:

```
<!-- This is a comment -->
```

Indentation Is Recommended

Although examples in this chapter indent SMIL tags to various levels to illustrate the SMIL structure, indentation is not required. Indenting your SMIL files as in the examples in this chapter will help you keep track of the SMIL functions, though.

Specifying Clip Locations

To add a clip to a presentation, you include in the SMIL body section a clip source tag that describes the clip type and location:

```
<audio src="rtsp://realserver.example.com:554/audio/first.rm"/>
```

There are several clip source tags, as described in the following table.

Clip Source Tags

Clip Tag	Used For
animation	Animation clips such as a Flash Player file (.swf).
audio	Audio clips such as RealAudio (.rm).
img	JPEG (.jpg), GIF (.gif), or PNG images (.png). See also "Defining Image Options" on page 123.
ref	Any clip type not covered by other attributes, such as a RealPix file (.rp).

(Table Page 1 of 2)

Clip Source Tags (continued)

Clip Tag	Used For
`text`	Static text clips (.txt).
`textstream`	Streaming RealText clips (.rt).
`video`	Video or other clips that display continuous motion, such as RealVideo (.rm).

(Table Page 2 of 2)

Within each clip source tag, a mandatory `src` attribute lists the clip location. How you specify this location depends on whether you will stream the presentation with RealServer, download the clip from a Web server, or play clips back from a local computer, as described in the following sections.

Note

The particular clip source tag you use does not affect clip playback because RealPlayer determines the actual clip type by other means. Specifying a video clip with an `<audio…/>` tag, for example, does not prevent RealPlayer from recognizing that the clip contains video. Although using a tag appropriate to the clip's contents helps you keep track of clips, you could specify all clips with `<ref…/>` tags, for example.

Linking to Clips on RealServer

When clips reside on RealServer, each clip's `src` attribute gives the clip's URL like this:

```
<audio src="rtsp://realserver.example.com:554/audio/first.rm"/>
```

The following table explains the URL components. Your RealServer administrator can give you the RealServer address, RTSP port, and directory structure.

RealServer URL Components

Component	Specifies
`rtsp://`	RTSP. Although RealServer also supports HTTP, streaming clips typically use RTSP.
`realserver.example.com`	RealServer address. This varies with each RealServer. It typically uses an identifier such as `realserver` instead of `www`. Or it may use a TCP/IP address (such as `172.2.16.230`) instead of a name.

(Table Page 1 of 2)

RealServer URL Components (continued)

Component	Specifies
:554	RealServer port for RTSP connections. Port 554 is the default, so you can leave this out of URLs unless the RealServer administrator chose a different port for RTSP communication. If the port number is required, separate it from the address with a colon.
/audio/	RealServer directory that holds the clip. The directory structure may be several levels deep. RealServer also uses "mount points" that invoke certain features, such as password authentication. Because these mount points appear to be directories in the URL, the request path does not mirror the actual directory path on the RealServer computer. The RealServer administrator can tell you the mount points and directories in the path.
first.rm	Clip file name.

(Table Page 2 of 2)

Additional Information

For more information on RTSP, see "RTSP and HTTP" on page 165.

Creating Relative URLs

If your presentation includes many clips that reside on the same server, you can make each URL relative to a base target that you define in the header. Here is an example:

```
<head>
  <meta name="base" content="rtsp://realserver.example.com/"/>
</head>
<body>
  <audio src="audio/first.rm"/>
  <audio src="audio/second.rm"/>
  <audio src="rtsp://realserver.real.com/media/third.rm"/>
</body>
```

Because the third clip in this example uses a full URL, the base target is ignored. For the first two clips, however, the src values are appended to the base target, effectively giving the clips the following URLs:

```
rtsp://realserver.example.com/audio/first.rm
rtsp://realserver.example.com/audio/second.rm
```

If no base target is given, RealPlayer assumes that the clip paths are relative to the location of the SMIL file. In the preceding example, therefore, you could leave the base target out if the SMIL file itself resides in a directory that contains the `audio` subdirectory that in turn holds the RealAudio clips. RealPlayer requests the clips with the same protocol it used to request the SMIL file.

Tip

The relative syntax for SMIL files works like relative links in HTML, so you can use directory notation such as "../". You can find additional information about this topic in an HTML reference guide.

Linking to Clips on a Web Server

To use a clip hosted on a Web server, use a standard HTTP URL such as the following in a clip source tag:

```
<img src="http://www.example.com/images/logo.gif"/>
```

URLs to clips on a Web server can also be relative to a base target, as described in the preceding section. Keep in mind that although a Web server can host any clip, a Web server cannot perform all the functions of RealServer.

Additional Information

See "Limitations on Web Server Playback" on page 171.

Linking to Local Clips

If your presentation clips will reside on the user's local computer, you need to include the SMIL file locally as well. The `src` attributes in the SMIL file list presentation clips in the following format:

```
src="audio/first.rm"
```

This example is a local *relative* link to a clip that resides one level below the SMIL file in the `audio` directory. For local access, you typically want to use relative links because you cannot be sure where users will place clips on their computers.

Alternately, you can use local *absolute* links to specify exact locations. The syntax for absolute links is the same as with HTML. It varies with operating systems, however, and you should be familiar with the directory syntax for the

system you are using. For example, the following absolute link syntax works for Windows computers, but not on UNIX or the Macintosh:

```
src="file://c:\audio\first.rm"
```

Warning

Microsoft Internet Explorer 3.0 version tries to display local SMIL files as HTML. To support this browser, start the presentation through a Ram file, as explained in "Creating a Ram File Manually" on page 173.

Grouping Clips

With the SMIL <seq> and <par> tags, you can create groups to structure your presentation. This section explains how to play clips in sequence or in parallel, as well as how to repeat groups or clips within groups.

Playing Clips in Sequence

Use the <seq> tag to play clips in sequence. In the following example, the second clip begins when the first clip finishes:

```
<seq>
  <audio src="audio/newsong.rm"/>
  <audio src="audio/oldsong.rm"/>
</seq>
```

If your presentation included just these clips, though, you would not need to use a <seq> tag. You could simply list the clips in order, and RealPlayer would play them in sequence. The <seq> tag is most commonly combined with <par> to create combinations of sequential and parallel groups.

Playing Clips in Parallel

You can play two or more clips at the same time by using the <par> tag. When RealServer streams parallel groups, it ensures that the clips stay synchronized. If some clip data does not arrive, RealServer either drops that data or halts the presentation playback until the data arrives. The following example combines a RealVideo clip with a RealText clip:

```
<par>
  <audio src="music/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

When multiple visual clips play in parallel, you need to define each clip's playback region. For more information, see "Laying Out Multiple Clips" on page 99. Also, be sure that clips playing in parallel do not exceed the audience connection's maximum bandwidth, which is described in "Audience Bandwidth Targets" on page 24.

Ending a Parallel Group on a Specific Clip

By default, a `<par>` group ends as soon as all clips finish playing back. You can modify this with an `end` attribute, as described in "Setting Begin and End Times" on page 92. Or you can use the `endsync` attribute to stop the group when a specific clip finishes playing:

```
<par endsync="first">
  ...
</par>
```

endsync="first"

Use the attribute `endsync="first"` to stop the `<par>` group when the clip with the shortest timeline finishes playing. All other clips in the group stop playing at that point, regardless of their playback status or any timing parameters specified for them.

endsync="last"

The attribute `endsync="last"` causes the `<par>` group to conclude when all clips have finished playing. Because this is the default value, you can omit the `endsync` attribute from the `<par>` tag to achieve this effect.

endsync="id(clip ID)"

The attribute `endsync="id(`*`clip ID`*`)"` causes the `<par>` group to conclude when a designated clip reaches its end. All other clips in the group stop playing at that point, regardless of their playback status or any timing parameters used with them. The designated clip must have a corresponding `id` value in its source tag:

```
<par endsync="id(vid)">
  <video id="vid" src="videos/newsong.rm" region="video"/>
  <textstream src="lyrics/newsong.rt" region="text"/>
</par>
```

Combining <seq> and <par> Tags

You can combine and nest <seq> and <par> tags as needed. The organization of these tags greatly affects the presentation playback, though. In the following example, clip 1 plays first. When it finishes, clip 2 and clip 3 play together. When both clip 2 and clip 3 have finished playing, clip 4 plays:

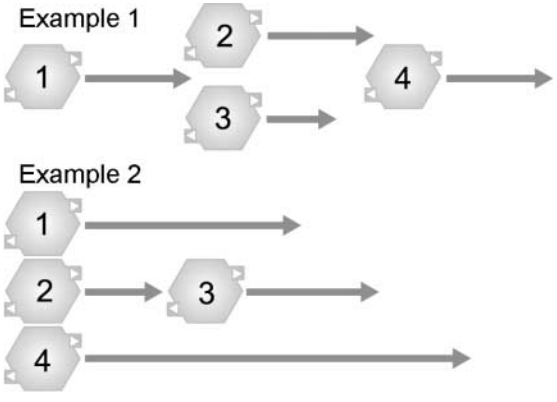
```
<seq>
   clip 1
   <par>
     clip 2
     clip 3
   </par>
   clip 4
</seq>
```

You get very different results, though, if you switch the <seq> and <par> groupings. In the next example, clips 1, 2, and 4 all begin at the same time. When clip 2 finishes, clip 3 starts:

```
<par>
   clip 1
   <seq>
     clip 2
     clip 3
   </seq>
   clip 4
</par>
```

The following illustration shows the difference between these groupings.

Different Playback Results with <seq> and <par> Groups



Specifying Timing

Using SMIL timing attributes, you can specify when a clip or group starts playing and how long it plays. All timing attributes are optional. If you do not use timing attributes, clips start and stop according to their normal timelines and their positions within `<par>` and `<seq>` groups. The easiest way to designate a time is with shorthand markers of `h`, `min`, `s`, and `ms`, as listed in the following table.

Timing Shorthand Markers and Examples

Timing Marker	Specifies	Example	Example Value
`h`	Hours	`2.5h`	2 hours, 30 minutes
`min`	Minutes	`2.75min`	2 minutes, 45 seconds
`s`	Seconds	`15.55s`	15 seconds, 550 milliseconds
`ms`	Milliseconds	`670.2ms`	670.2 milliseconds

Tip

Decimal values are not required. You can express two seconds as `"2s"` or `"2.0s"`, for example.

Normal Play Time Format

You can also express time attributes in a `hh:mm:ss.xy` format. Here, `hh` is hours, `mm` is minutes, `ss` is seconds, `x` is tenths of seconds, and `y` is hundredths of seconds. In this example:

```
begin="02:34.0"
```

the time value is 2 minutes, 34 seconds. If the value does not include a decimal point, RealPlayer interprets the last value as seconds. For example, RealPlayer would read the following value as 2 minutes, 34 seconds rather than as 2 hours, 34 minutes:

```
begin="02:34"
```

Setting Begin and End Times

The `begin` attribute works for any clip or group. You can use it to start a clip at a specific point within a presentation's timeline:

```
<video src="videos/newsong.rm" begin="20.5s"/>
```

Were this clip in a `<par>` group, the `begin` attribute would start the clip playing back 20.5 seconds after the group becomes active. If the clip were in a `<seq>`

group, the attribute would delay the clip's normal playback by inserting 20.5 seconds of blank time before the clip starts. Hence, timing is relative to the start of the <seq> or <par> group, not the start of the overall presentation.

Additionally, you can set an end attribute alone or in combination with a begin attribute as shown here:

```
<video src="videos/newsong.rm" begin="20.5s" end="62.7s"/>
```

In this example, the clip ends 62.7 seconds into its part of the presentation timeline, playing a total of 42.2 seconds regardless of the length of its internal timeline. If the video's internal timeline is shorter (30 seconds, for example), the fill attribute described in "Setting a Fill" on page 95 determines what happens on-screen after the video stops playing but before the end time is reached.

Using Begin and End Times with Groups

You can use the begin or end attributes with a <par> or <seq> group:

```
<par begin= "5s" end= "3.5min">
  ...
</par>
```

This begin value delays group playback until 5 seconds after the preceding group or clip finishes. The end time sets all clips in the group to stop playing after 3.5 minutes, regardless of their playback statuses. If all clips reach their normal conclusions by 3 minutes and 20 seconds after the group starts, for example, the next group or clip starts after 10 seconds of blank playback time.

Warning

Do not set an end time along with an endsync attribute in a <par> group, as this sets up conflicting end times. For more on endsync, see "Ending a Parallel Group on a Specific Clip" on page 90.

Setting Internal Clip Begin and End Times

The clip-begin and clip-end attributes specify a clip's internal timing marks where playback begins and ends. You can use them with clips that have internal timelines, such as audio, video, and animation. Do not use them with groups or static clips such as still images. The following example is for a video clip:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s"/>
```

Here, the clip starts playing at its internal 10.5-second mark rather than at its actual beginning. It stops when it reaches its 50.7-second mark, having played for a total of 40.2 seconds.

Note

Do not use clip-begin and clip-end for a live broadcast or when playing clips back from a Web server. For more information, see "Limitations on Web Server Playback" on page 171 and "Using SMIL with a Broadcast" on page 186.

Combining clip-begin and clip-end with the begin and end Attributes

You can combine clip-begin and clip-end attributes with begin and end attributes. Here, a begin time has been added to the preceding example:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s" begin="5s"/>
```

The begin time delays the clip's normal starting point by 5 seconds. When this time elapses, the clip starts at its 10.5-second internal timeline marker and plays for 40.2 seconds. In this case, the clip-end attribute determines how long the video is active. But you could also add an end attribute to modify this behavior, as shown in the following example:

```
<video src="videos/newsong.rm" clip-begin="10.5s" clip-end="50.7s" begin="5s"
  end="50s"/>
```

Combined with the begin value, the end value of 50 means that the clip's "window of activity" within the presentation is 45 seconds. Because the clip stops playing after 40.2 seconds, there is an extra 4.8 seconds during which the clip does not play but remains active. How the video window appears during these final seconds depends on the fill attribute, as described in "Setting a Fill" on page 95.

Setting Durations

The dur attribute controls how long a clip or group stays active after it starts to play back. You may want to use it in place of the end attribute. For example, in the following clip:

```
<video src="videos/newsong.rm" begin="20.5s" end="62.7s"/>
```

you can substitute a dur attribute for the end attribute to achieve the same result:

```
<video src="videos/newsong.rm" begin="20.5s" dur="42.2s"/>
```

In both examples, the clips stop 42.2 seconds after starting to play back. With the `end` attribute, the total playing time is the `end` value minus the `begin` value. The `dur` attribute ignores the `begin` value, stopping the clip 42.2 seconds after it starts. Use either `end` or `dur`, therefore, depending on how you want to measure time. Do not use both attributes in the same tag, however.

Note

The `dur` attribute can function like `end` in a group, too. For more information, see "Using Begin and End Times with Groups" on page 93.

The `dur` attribute is useful for determining how long graphic images are displayed. In the following example, the image disappears 14.5 seconds after it appears:

```
<img src="graphics/poster.jpg" dur="14.5s"/>
```

Indefinite Durations

In RealPlayer 8 or later, you can use `dur="indefinite"` with a clip to make the presentation appear to be a live broadcast. Although a visual clip stops playing at its normal end or its `clip-end` time, its last frame stays frozen on-screen while the presentation timeline continues to advance. As with a live broadcast, the RealPlayer timeline slider is disabled, so viewers cannot seek through the presentation.

Tip

For image clips in `<par>` groups, use `fill="freeze"` instead of `dur="indefinite"` to display the image on-screen as long as the group is active.

Setting a Fill

Use the `fill` attribute to determine what happens to a given clip immediately after it plays to its actual ending or its specified end time, or as soon as its duration elapses. The `fill` value can be either `remove` or `freeze`.

fill="remove"

The default value `fill="remove"` removes the clip. When this attribute is used with a still image, the image disappears as soon as the end time has been reached.

fill="freeze"

Use fill="freeze" to freeze a clip on its last frame. When you use fill="freeze" with a video, the video's last frame stays on the screen. Suppose that you have a 20-second video and specify a 30-second duration with freeze:

```
<video src="videos/newsong.rm" dur="30s" fill="freeze"/>
```

When the video finishes playing, its last frame displays for an additional 10 seconds. The video disappears when the end time is reached. The fill="freeze" attribute has no effect on audio. Do not use fill="freeze" for a graphic image that also uses a dur attribute.

Clip Timing Example

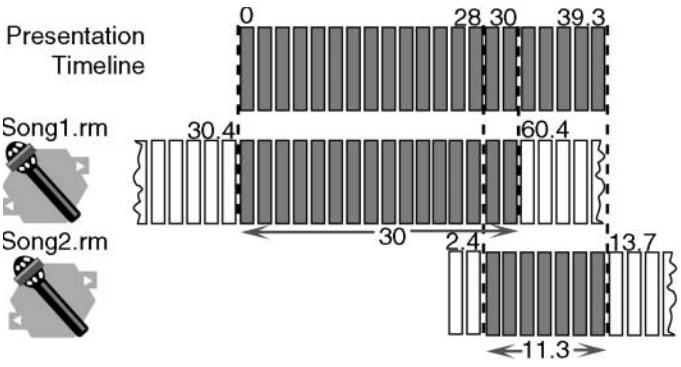
The following example shows two audio clips with different timing options:

```
<par>
  <audio src="song1.rm" clip-begin="30.4s" dur="30s"/>
  <audio src="song2.rm" begin="28s" clip-begin="2.4s" clip-end="13.7s"/>
</par>
```

The timing options modify the <par> tag so that the two clips start at different times. The first clip begins to play immediately, but starts at 30.4 seconds into its timeline, playing for exactly 30 seconds.

The second clip is delayed for 28 seconds. That means it overlaps the first clip by 2 seconds. It starts at 2.4 seconds into its timeline and ends at 13.7 seconds into its timeline, playing for 11.3 seconds. The total playing time for this group is 30 seconds for the first clip, plus 11.3 seconds for the second clip, minus the 2 second overlap: 39.3 seconds. The following illustration shows how the clip timelines relate to the overall presentation timeline.

Clip Timing Example



Repeating a Clip or Group

A repeat attribute in a clip source tag or group tag repeats the clip's or group's playback. You can specify the number of times that the clip or group plays, or you can make the clip or group repeat indefinitely. During each playback, RealPlayer "prebuffers" each subsequent repetition to keep the presentation from halting when the clip replays. Note, however, that when you stream a clip with RTSP or HTTP, each repetition consumes bandwidth because RealPlayer does not cache the clip.

Tip

If you use CHTTP to cache a repeating clip on RealPlayer, the clip consumes bandwidth only the first time it plays. The clip must be smaller than RealPlayer's cache size of 4 MB, however. For more information, see "Caching Files on RealPlayer" on page 132.

Repeating Playback a Specific Number of Times

The repeat="*n*" attribute plays a clip or group a specified number of times. You can add it to a clip source tag, as in the following example:

```
<video src="videos/newsong.rm" repeat="4"/>
```

An end or a dur attribute can modify clip repetition. Suppose that the following video has a 3-minute internal timeline. The end and fill attributes freeze the clip on-screen for 30 seconds before playback repeats:

```
<video src="videos/newsong.rm" repeat="4" end="3.5min" fill="freeze"/>
```

Repeating Clips in Groups

When adding a repeating clip to a group, pay close attention to how the clip repetition affects the group timeline. In the following example, an audio clip within a <par> group plays twice. Assuming that the audio clip lasts longer than the text stream, the <par> group ends when the audio clip concludes its second playback:

```
<par>
  <audio src="music/newsong.rm" repeat="2"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

In the following <seq> group, the second video plays only after the first video plays twice:

```
<seq>
  <video src="videos/newsong.rm" repeat="2"/>
  <video src="videos/newsong2.rm"/>
</seq>
```

Repeating Groups of Clips

You can add a repeat="*n*" attribute to a <seq> or <par> tag to make the entire group repeat. As with an individual clip, including an end or dur attribute causes repetition to occur only when the specified time is reached. In the following example, the group repeats after 4 minutes, regardless of where the clips are in their timelines:

```
<par repeat="2" dur="4min">
  <audio src="music/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

Looping Playback Indefinitely

The repeat="indefinite" attribute causes a clip or group to repeat until another SMIL attribute or user event stops the playback loop. It works only with RealPlayer 8 or later, and has no effect on earlier RealPlayers. In the following example, the audio clips repeats continuously until the viewer clicks the RealPlayer **Stop** button:

```
<audio src="music/newsong.rm" repeat="indefinite"/>
```

You can specify how long each loop lasts by using an end or dur attribute. In the next example, each loop lasts for 1.5 minutes:

```
<audio src="music/newsong.rm" end="1.5min" repeat="indefinite"/>
```

You can also use repeat="indefinite" in a <seq> or <par> tag. In the following example, the first video plays again every time the third video stops:

```
<seq repeat="indefinite">
  <video src="videos/newsong.rm"/>
  <video src="videos/newsong2.rm"/>
  <video src="videos/newsong3.rm"/>
</seq>
```

Looping Clips in Groups

In a sequence of clips, using repeat="indefinite" for a clip prevents the sequence from ending unless the <seq> tag itself specifies the end time. In the following

example, the first two clips play, and then the last clip loops indefinitely until the 30-minute sequence duration elapses:

```
<seq dur="30min">
  <video src="videos/newsong.rm"/>
  <video src="videos/newsong2.rm"/>
  <video src="videos/newsong3.rm" repeat="indefinite"/>
</seq>
```

In a <par> group, you can use dur to end playback at a specified time, or you can include endsync="id(*clip ID*)" to stop the group when a clip other than the looping clip finishes. In the following example, the audio loop stops when the RealPix slideshow concludes:

```
<par endsync="id(pix)">
  <audio src="music/background.rm" repeat="indefinite"/>
  <ref src="pictures/promo.rp" id="pix"/>
</par>
```

Additional Information

See "Ending a Parallel Group on a Specific Clip" on page 90 for more information on endsync.

Laying Out Multiple Clips

If your presentation plays only one clip at a time, you do not need to create a layout. Each clip automatically plays in the main RealPlayer window, the window resizing automatically for each new clip. If you want to keep the playback area the same size as successive clips play, or if your presentation displays several clips at a time, you need to define SMIL regions.

➤ To define SMIL regions:

1. In the SMIL file header, define the layout between <layout> and </layout> tags.
2. In the layout section, define a root-layout region that sets the overall size of the RealPlayer main window.
3. Define the sizes and locations of any number of named playback regions.
4. In the SMIL file body, use region attributes to assign source clips to the named playback regions.

A SMIL file that includes a layout therefore takes the following form:

```
<smil>
  <head>
    <layout>
      <root-layout ...defines the overall window size... />
      <region id="name1" ...defines a named playback region within root-layout... />
      <region id="name2" ...defines a named playback region within root-layout... />
      ...
    </layout>
  </head>
  <body>
    <ref src="..." region="name1" ...assigns a clip to a region by name... />
    <ref src="..." region="name2" ...assigns a clip to a region by name... />
    ...
  </body>
</smil>
```

Additional Information

For instructions on using RealPlayer's Netscape plug-in or ActiveX control to lay out a presentation in a Web page instead of in RealPlayer, see "Laying Out SMIL Presentations" on page 153.

Creating the Root Layout Region

Using the <root-layout…/> tag, you specify the width and height of the overall playback area in pixels. The following example creates a root-layout region 250 pixels wide by 230 pixels high. When the presentation begins, the RealPlayer window expands to this size. Other regions measure their top and left offsets from the upper-left corner of this root-layout region:

```
<head>
  <layout>
    <root-layout width="250" height="230"/>
    ...other regions defined after root-layout...
  </layout>
</head>
```

Note

Although you can omit root-layout to have RealPlayer calculate the playback area based on the sizes of the other regions, it is better to define root-layout to avoid unexpected results. Keep in mind that you cannot play clips in the root-layout region.

Defining Playback Regions

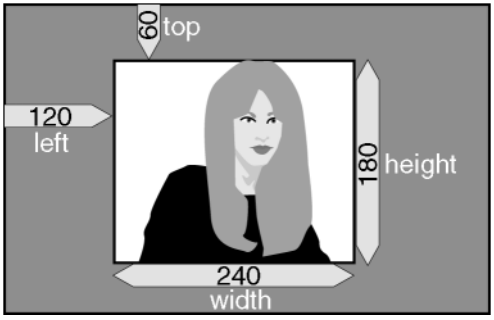
You create playback regions for clips by defining <region> tags in the layout section. All playback regions must lay within the root-layout region. Any part of a playback region that lays outside the root-layout region is cut off. You define the playback regions by using a simple coordinate system measured across and down from the upper-left corner of the root-layout region. All measurements are in pixels or percentages, with zero pixels as the default. Each <region…/> tag must define top, left, width, and height attributes, as described in the following table.

Region Size and Placement Attributes

Attribute	Sets	Pixel Example	Percentage Example
top	Offset from top of root-layout	top="60"	top="10%"
left	Offset from left side of root-layout	left="120"	left="20%"
width	Region width	width="240"	width="40%"
height	Region height	height="180"	height="30%"

The following illustration shows a playback region laid out within a root-layout region, based on the pixel measurement examples in the preceding table.

Basic Region Layout



In addition to its region measurements, each playback region must have an id="*name*" attribute that identifies it. You assign clips to a region based on the region's name. The following SMIL example defines two regions named "videoregion" and "textregion":

```
<head>
  <layout>
    <root-layout background-color="maroon" width="250" height="230"/>
    <region id="videoregion" top="5" left="5" width="240" height="180"/>
    <region id="textregion" top="200" left="5" width="240" height="20"/>
  </layout>
</head>
```

In this example, both regions are offset 5 pixels to the right of the root-layout region's left edge. The video region displays 5 pixels down from the top of the root-layout region, and the text region displays 200 pixels down. The following illustration shows this placement.

SMIL Sample Layout with Video and Text Windows

Using Percentages for Region Layout

For a region's height, width, and offset measurements, you can use percentages that reflect a fraction of the root-layout region's size. The following example uses percentages to define playback areas similar to those defined in the preceding example:

```
<head>
  <layout>
    <root-layout background-color="maroon" width="250" height="230"/>
    <region id="videoregion" top="2%" left="2%" width="96%" height="78%"/>
    <region id="textregion" top="80%" left="2%" width="96%" height="18%"/>
  </layout>
</head>
```

Tips for Defining Regions

Note the following about SMIL regions:

- All regions used in a presentation must be defined in the header. All regions are created at the beginning of the presentation, so regions containing background colors as described in "Adding Background Colors" on page 103 may obscure other regions they overlap.
- Using region transparency and the `z-index` attribute, you can emulate dynamic creation and destruction of regions. For details, see "Hiding Regions with z-index" on page 207.
- To use a graphic such as a GIF, JPEG, or PNG image as a background, define a playback region the same size as the root-layout region. Display the image in this region, and use the `z-index` attribute to have it displayed behind other regions. For more information, see "Ordering Overlapping Regions with z-index" on page 106.

Keep in mind the following when using percentage values to define regions:

- You must include a root-layout region defined in pixels when specifying region measurements in percentages.
- You can mix pixels and percentages. You could define the `top` and `left` offset measurements in percentages, for example, while specifying the `width` and `height` measurements in pixels.
- You can use both whole and decimal values for percentages. For example, the values "4%" and "4.5%" are both valid.
- Because a region is clipped at the boundary of the root-layout region, no percentage value can effectively be more than 100%.

Adding Background Colors

By default, the root-layout region is black. All other regions use transparency as their default. In the SMIL layout, you can specify a different background color for any region, as in the following example:

```
<layout>
  <root-layout background-color="maroon"/>
  <region id="videoregion" background-color="silver".../>
  <region id="textregion" background-color="#C2EBD7".../>
</layout>
```

For the color value, use any RGB hexadecimal value (#RRGGBB) supported by HTML, or one of the following predefined color names, listed here with their corresponding hexadecimal values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuchsia (#FF00FF)	red (#FF0000)	maroon (#800000)
lime (#00FF00)	olive (#808000)	green (#008000)	purple (#800080)
aqua (#00FFFF)	teal (#008080)	blue (#0000FF)	navy (#000080)

Transparency in SMIL Regions

A SMIL region is transparent if you do not define its background color. You can also explicitly specify "transparent" as its background color. Transparency means that the region is not visible until a clip starts to play in it. Note, however, that if a clip that contains transparency (such as a GIF image) plays in a transparent region, you cannot see through the clip's transparent areas to underlying regions or clips.

Defining How Clips Fit Regions

When a clip is encoded at a size different from the playback region's defined size, the `<region.../>` tag's `fit` attribute determines how the clip fits the region:

```
<region id="videoregion" width="128" height="64" fit="meet"/>
```

The `fit` attribute uses one of the values described in the following table. If you do not specify a `fit` attribute, the clip uses the default value `hidden`. In no case will a clip display outside the boundaries of its region.

Region Fit Attributes

Attribute	Action
`fill`	Scale the clip so that it fills the region exactly. Image distortion occurs if the encoded clip and playback region have different height/width ratios.
`hidden` (default)	Keep the clip at its encoded size, and place it in the region's upper-left corner. If the clip is smaller than the region, fill remaining space with the region's background color. If the clip is larger than the region, crop out the area that does not fit.
`meet`	Place the clip at the region's upper-left corner. Scale the clip and preserve its height/width ratio until one dimension is equal to the region's size and the other dimension is within the region's boundaries. Fill empty space with the region's background color.

(Table Page 1 of 2)

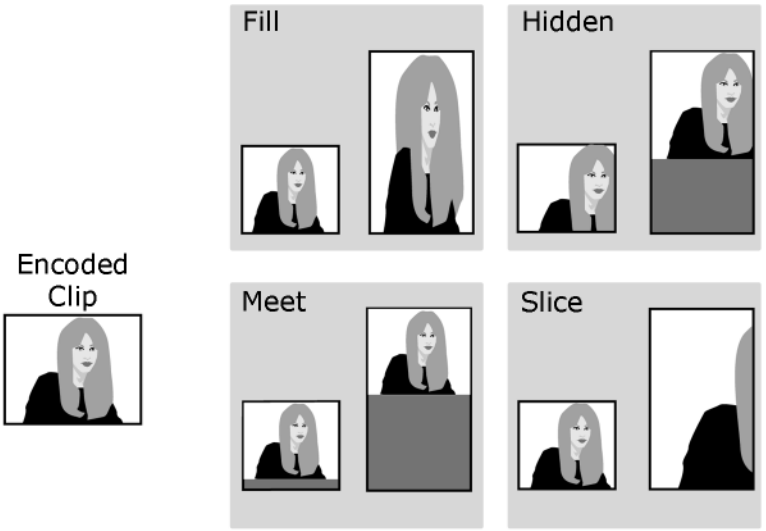
Region Fit Attributes (continued)

Attribute	Action
`scroll`	Place the clip at the region's upper-left corner. Display the clip as its normal size, adding horizontal or vertical scroll bars if the clip extends beyond the region's boundaries. (RealPlayers earlier than RealPlayer 8 display clips as the default "`hidden`" instead.)
`slice`	Place the clip at the region's upper-left corner. Scale the clip and preserve its height/width ratio until one dimension is equal to the region's size and the other dimension overflows the region's boundaries. Crop the overflow.

(Table Page 2 of 2)

The following illustration shows the effects that particular fit attributes have on a source clip played in windows with different sizes and aspect ratios.

Different Clip Scaling Results Based on the fit Attribute



Tip

When scaling clips inside a region, keep in mind that different types of media scale with different results. A video scaled larger than its encoded size may not look good. Vector-based media such as Flash animation, on the other hand, scale more easily to fit different region sizes. Also, remember that scaling a clip consumes CPU power on the RealPlayer computer.

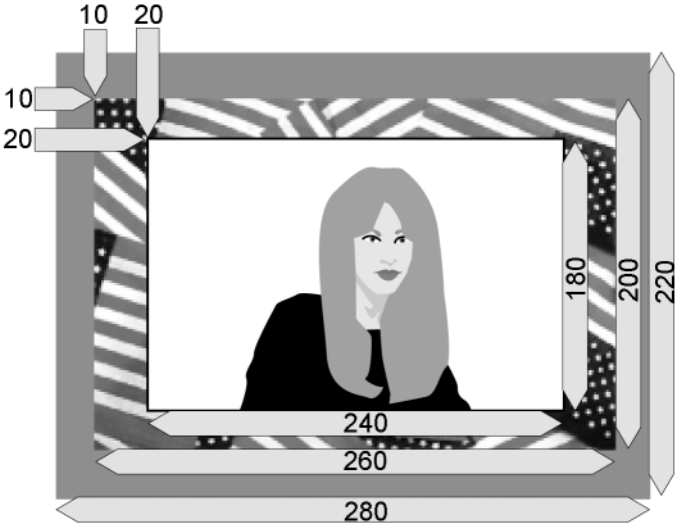
Ordering Overlapping Regions with z-index

If regions overlap, you can use thez-index attribute in <region…/> tags to determine which regions appear in front. The following example creates a video region that overlaps an image region:

```
<layout>
  <root-layout background-color="gray" width="280" height="220"/>
  <region id="image" top="10" left="10" width="260" height="200" z-index="0"/>
  <region id="video" top="20" left="20" width="240" height="180" z-index="1"/>
</layout>
```

This example defines a gray root-layout region 220 pixels high by 280 pixels wide. A smaller image region is centered within this gray background. Its z-index value of 0 makes it display behind all other regions, but not behind the root-layout region. The video region centered in the image region appears on top of that region because of its higher value for z-index. You could have another region overlap the video region by setting z-index to 2, 5, or 29, for instance. The following illustration shows these regions.

Regions Overlapping Through z-index



Tips for Defining Z-Index Values

The following are points to observe when using z-index:

- The root-layout region is always behind all other regions and does not use z-index.

- The z-index values can include negative integers (such as -4), 0 (zero), and positive integers (such as 5). A region with a z-index value of -4, for example, displays behind a region with a value of 0, which displays behind a region with a value of 5.
- The default value of 0 (zero) applies if you don't specify z-index.
- Using strictly sequential values such as 0, 1, 2, 3, 4 helps you keep track of the layers, but is not necessary. A sequence such as 0, 1, 6, 19, 34 works just as well, and leaving gaps in the sequence makes it easier to insert layers later.
- Nonoverlapping clips can have the same values. Side-by-side videos can both have z-index="3", for example.
- When overlapping clips have the same value, the clip that starts later in the presentation appears in front.

Additional Information

See "Hiding Regions with z-index" on page 207 for information on using z-index to make regions appear to come and go dynamically.

Assigning Clips to Regions

After you define the SMIL layout in the header section, you use region attributes within source tags to assign each clip to a region based on the region's id attribute value. In the following example, the video and text clips are assigned to the video and text regions defined in the header:

```
<smil>
  <head>
    <layout>
      <root-layout background-color="maroon" width="250" height="230"/>
      <region id="videoregion" top="5" left="5" width="240" height="180"/>
      <region id="textregion" top="200" left="5" width="240" height="20"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="video.rm" region="videoregion"/>
      <audio src="audio.rm"/>
```

```
    <textstream src="text.rt" region="textregion"/>
   </par>
  </body>
</smil>
```

You can reuse regions by assigning sequential clips to them. For example, you can play a video clip in a region, then display another clip in that region after the first clip finishes. Don't assign the same region to two clips that play at the same time, however. You don't assign audio clips to regions at all because audio does display on the screen.

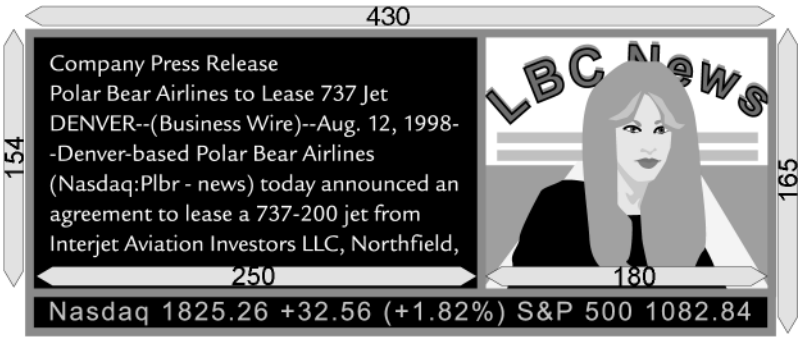
SMIL Layout Example

The following example displays three regions: a news region, a video region, and a stock ticker region. The news and video regions are arranged side by side at the top of the RealPlayer display window. The stock ticker region appears below them.

```
<smil>
 <head>
  <!--presentation with 2 text clips and 1 video clip-->
  <meta name="title" content="News of the Week"/>
  <layout>
   <root-layout width="430" height="165"/>
   <region id="newsregion" top="0" left="0" width="250" height="144"/>
   <region id="videoregion" top="0" left="250" width="180" height="144"/>
   <region id="stockregion" top="145" left="0" width="430" height="20"/>
  </layout>
 </head>
 <body>
  <par>
   <!--play these 3 clips simultaneously-->
   <textstream src="news.rt" region="newsregion"/>
   <video src="newsvid.rm" region="videoregion"/>
   <textstream src="stocks.rt" region="stockregion"/>
  </par>
 </body>
</smil>
```

The following illustration shows the design of these regions.

SMIL Sample Layout with Text, Video, and Stock Ticker Windows



Switching Between Alternate Choices

With the <switch> tag, you can specify multiple options that RealPlayer can choose between. The <switch> group specifies any number of choices in this form:

```
<switch>
  <choice1 test-attribute="value1"/>
  <choice2 test-attribute="value2"/>
  …
</switch>
```

RealPlayer looks at choices in order, evaluating each test attribute and its value to determine which clip to choose. The choices are typically clip source tags such as <video *test-attribute="value"*/>, but RealPlayer can also choose between groups when test attributes appear in <par> or <seq> tags.

Note

Every RealPlayer must have a viable choice within a <switch> statement. RealPlayer will not play any clip in the <switch> group if it finds no satisfactory test attribute. The next sections explain how to guarantee that every RealPlayer has a viable option.

Setting Language Choices

When source clips are in different languages, use a <switch> test attribute of system-language. The following example shows a video slideshow with separate audio narrations in French, German, Spanish, Portuguese, Japanese, Korean, and English. Each RealPlayer would request the same slideshow (seattle.rp),

but choose an audio clip based on its language preference and its evaluation of the system-language codes:

```
<par>
  <ref src="slides/seattle.rp"/>
  <!-- select audio based on RealPlayer language preference setting -->
  <switch>
    <audio src="french/seattle.rm" system-language="fr"/>
    <audio src="german/seattle.rm" system-language="de"/>
    <audio src="spanish/seattle.rm" system-language="es"/>
    <audio src="portuguese/seattle.rm" system-language="pt"/>
    <audio src="japanese/seattle.rm" system-language="ja"/>
    <audio src="korean/seattle.rm" system-language="ko"/>
    <audio src="english/seattle.rm"/>
  </switch>
</par>
```

Because the last option does not have a test attribute, a RealPlayer that does not have French, German, Spanish, Portuguese, Japanese, or Korean explicitly set as its preferred language chooses the English clip, regardless of its actual language setting. RealPlayer evaluates options in order, so the last option should be, as shown in this example, a default language that applies if no other option is viable. If you list no default option and certain RealPlayers prefer languages other than the ones you list, those RealPlayers will not play any of the clips.

Additional Information

Appendix E lists the system-language codes such as "fr" you use to designate content in different languages.

Setting Bandwidth Choices

To stream different clips to viewers at different connection speeds, use the <switch> tag to define options each RealPlayer can choose based on its connection bandwidth. Within the <switch> tag, system-bitrate attributes for clips or groups give the approximate bits per second each clip or group uses. The following sample <switch> tag lists two different RealPix presentations. The first is for connections that have at least 80 Kbps of bandwidth. The second is for slower connections, down to 28.8 Kbps modems:

```
<switch>
  <ref src="pix/slides1.rp" system-bitrate="80000"/>
  <ref src="pix/slides2.rp" system-bitrate="20000"/>
</switch>
```

As shown above, list the bandwidth choices from fastest to slowest. RealPlayer evaluates options in order, selecting the first option it can play. If the 20,000 bps option were first, a RealPlayer with a high-speed connection would choose it because it is the first viable option. Also ensure that the last option satisfies the slowest connection speed you want to support. If the last choice is `system-bitrate="60000"`, for example, RealPlayers on modems will not play the presentation because its bandwidth requirement is too high.

The more complex example below shows three sets of clips. Each `<par>` tag has a `system-bitrate` attribute that lists the approximate bandwidth the clips as a whole consume. Note that each group uses the same RealText clip, but has different RealAudio and RealPix clips created for its bandwidth:

```
<switch>
  <par system-bitrate="225000">
    <!--RealPlayers with 225 Kbps or faster connections will choose this group-->
    <audio src="audio/music1.rm"/>
    <ref src="pix/slides1.rp" region="images"/>
    <textstream src="words/narration.rt" region="text"/>
  </par>
  <par system-bitrate="80000">
    <!--RealPlayers with connections between 80 and 225 Kbps get this group-->
    <audio src="audio/music2.rm"/>
    <ref src="pix/slides2.rp" region="images"/>
    <textstream src="words/narration.rt" region="text"/>
  </par>
  <par system-bitrate="20000">
    <!--RealPlayers with connections between 20 and 80 Kbps get this group-->
    <audio src="audio/music3.rm"/>
    <ref src="pix/slides3.rp" region="images"/>
    <textstream src="words/narration.rt" region="text"/>
  </par>
</switch>
```

Additional Information

The table "Maximum Streaming Rates" on page 24 gives bandwidth guidelines for various network connections. "Writing Complex SMIL Switch Statements" on page 209 gives several examples of `<switch>` statements.

Linking to Other Media

A SMIL file can define links to other media. A video might link to a second video, for example. When the viewer clicks the link, the second video replaces the first. Or the video could link to an HTML page that opens in the viewer's browser. You can even define areas as hot spots with links that vary over time. The bottom corner of a video can link to a different URL every ten seconds, for instance.

Note

Some clips can also define hyperlinks. A RealText clip, for example, can define hyperlinks for portions of text. When a viewer clicks an area where a clip link and a SMIL file link overlap, the SMIL link is used.

Making a Source Clip a Link

The simplest type of link connects an entire source clip to another clip. As in HTML, you define the link with `<a>` and `</a>` tags. But whereas you enclose text between `<a>` and `</a>` in HTML, you enclose a clip source tag between `<a>` and `</a>` in SMIL:

```
<a href="rtsp://realserver.example.com/video2.rm">
  <video src="video.rm" region="videoregion"/>
</a>
```

This example links the source clip `video.rm` to the target clip `video2.rm`. When a viewer moves the cursor over the source clip as it plays, the cursor turns to a hand icon to indicate that the clip is a link. When the viewer clicks `video.rm` as it plays, `video2.rm` replaces it. The URL begins with `rtsp://` if the linked clip streams to RealPlayer from RealServer, or `http://` if the file downloads to the browser from a Web server. When targeting a browser, include the `show` attribute as described below.

Targeting RealPlayer or a Browser

An `<a>` tag or `<anchor>` tag (see "Defining Hot Spot Links" on page 113) can include a `show` attribute that determines where a linked clip displays:

```
<a href="http://www.example.com/index.htm" show="new">
  <video src="video.rm" region="videoregion"/>
</a>
```

show="replace"

The default attribute `show="replace"` causes the linked clip to replace the source clip in RealPlayer. This default behavior also occurs if you do not include the `show` attribute in the link. The following are important differences between RealPlayer and Web browsers to keep in mind when creating links:

- RealPlayer does not include a **Back** button that allows the viewer to return to the link source clip after clicking the link.
- Clicking a link to another SMIL file or clip removes any existing regions. If you have three regions defined and the viewer clicks a link in one region, for example, the target clip replaces all clips in all regions. You can preserve regions by targeting a SMIL file that defines the same set of regions. You cannot preserve the timeline positions of clips playing in those regions when the viewer clicks the link, however.

Tip

RealNetworks has created an extension to SMIL that lets you open a linked presentation in a new RealPlayer window. For more information, see "Popping Up New RealPlayer Windows" on page 127.

Additional Information

"Linking to a SMIL File" on page 116.

show="new" and show="pause"

The values `new` and `pause` both open the linked clip in the viewer's default browser. The source clip continues to play in RealPlayer if you use `show="new"`. With `show="pause"`, the source clip pauses in RealPlayer. The viewer can restart playback at any time, though, by clicking RealPlayer's **Play** button.

Use either `show="new"` or `show="pause"` to open a Web page or another clip viewable within a browser. You can use these attributes to link a RealSystem presentation to your home page, for example. Do not use them to link to another clip played in RealPlayer, however, such as a SMIL file or a RealVideo clip.

Defining Hot Spot Links

Within a SMIL file you can define hot spots using an `<anchor>` tag. Whereas the `<a>` tag turns the entire source clip into a link, the `<anchor>` tag turns only

a defined area into a link. With <anchor> tags you can create links similar to those in HTML image maps. But SMIL links can be temporal as well as spatial. A link might be valid for just ten seconds during a source clip's timeline, for instance.

Setting an Anchor

The <anchor> tag differs from the <a> tag in that you place it within the clip source tag rather than before it:

```
<video src="video.rm" region="videoregion">
  <anchor href="rtsp://realserver.example.com/video2.rm" .../>
</video>
```

An <anchor> tag ends with a closing slash. But the clip source tag does not end with a closing slash as it normally would. Instead, the source tag and its subsequent <anchor> tags are followed by a closing source tag, such as </video>. The <anchor> tag includes an href attribute that uses rtsp:// if the linked clip streams from RealServer, or http:// if the file downloads to the browser from a Web server

Additional Information

For information on RTSP URLs, see "Linking to Clips on RealServer" on page 86. To target a browser with a link, see "Targeting RealPlayer or a Browser" on page 112.

Defining Spatial Coordinates

The <anchor> tag's coords attribute defines spatial coordinates for the hot spot rectangle. Coordinate values in pixels or percentages define the rectangle's offset from the upper-left corner of the source clip as shown in this example:

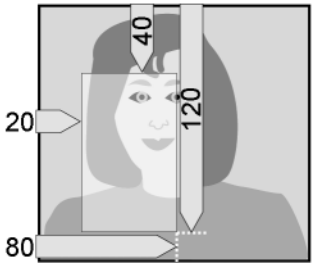
```
<video src="video.rm" region="videoregion">
  <anchor href="..." coords="20,40,80,120"/>
</video>
```

The coordinate values for the hot spot rectangle follow this order:

1. left side pixel or percentage value
2. top pixel or percentage value
3. right side pixel or percentage value
4. bottom pixel or percentage value

The preceding example uses pixel values to define a hot spot 60 pixels wide (80 pixels minus 20 pixels) 80 pixels high (120 pixels minus 40 pixels). It produces a hot spot like that shown in the following illustration.

Hot Spot (Hyperlink) Defined with SMIL



Percentage Values

The `coords` attribute can also use percentage values to create hot spots relative to the source clip's size. The following sample places in the center of the source clip a hot spot that is a quarter the size of the source clip:

```
<video src="video.rm" region="videoregion">
  <anchor href="..." coords="25%,25%,75%,75%"/>
</video>
```

The following table lists sample percentage coordinates that define hot spots for a source clip. Each hot spot is a quarter the size of the source clip.

Sample Hot Spot Percentage Coordinates

Hot Spot Rectangle Position	Coordinate Attribute
Upper-left quadrant	`coords="0,0,50%,50%"`
Upper-right quadrant	`coords="50%,0,100%,50%"`
Lower-left quadrant	`coords="0,50%,50%,100%"`
Lower-right quadrant	`coords="50%,50%,100%,100%"`
Center	`coords="25%,25%,75%,75%"`

Tips for Defining Anchor Coordinates

Note the following when defining hot spots:

- You can mix pixels and percentages. For example, the coordinates "50,50,100%,100%" place the hot spot's left and top boundaries in and down 50 pixels from the source clip's upper-left corner, respectively. But

the hot spot's right and bottom boundaries extend to the source clip's right and bottom edges, respectively, no matter the source clip's size.

- Values such as "30,30,10,10" are ignored. Here, the hot spot's left side is defined as farther to the right than its right side. As well, the top is defined to be below the bottom.
- You can use whole and decimal values for percentages. For example, the values "4%" and "4.5%" are both valid.
- A hot spot defined to extend beyond the source clip is clipped at the clip's edge. For example, if the hot spot has coordinates "50,50,300,300" but the source clip is 200 by 200 pixels, the hot spot's effective coordinates are "50,50,200,200". For this reason, no percentage value can effectively be more than 100%.

Setting Temporal Coordinates

In addition to defining spatial coordinates, the `<anchor>` tag can set temporal attributes that specify when the link is active. If you do not include temporal attributes, the link stays active as long as the source clip appears on-screen. To add timing attributes, use the SMIL `begin` and `end` values. (You cannot use `dur`, `clip-begin`, or `clip-end`.)

The following example creates two temporal links for the clip `video.rm`. The first link is active for the first five seconds of playback. The second link is active for the next five seconds. Because no spatial coordinates are given, the entire video is a link:

```
<video src="video.rm" region="videoregion">
  <anchor href="rtsp://.../video2.rm" begin="0s" end="5s"/>
  <anchor href="rtsp://.../video3.rm" begin="5s" end="10s"/>
</video>
```

Additional Information

See "Setting Begin and End Times" on page 92. The attributes use the SMIL timing values described in "Specifying Timing" on page 92.

Linking to a SMIL File

A SMIL file can define a link to another SMIL file or another part of itself. For example, a video played through a SMIL file may link to another SMIL file so that when a viewer clicks the video, a new presentation starts up in RealPlayer.

To do this, you simply set the href attribute for the <a> or <anchor> tag to the new SMIL file's URL.

You can also link to portions of a SMIL file. The following example from a target SMIL file uses id attributes (such as those used in regions to create region names) to define a target name for a <par> tag that groups a video and a text clip. This id attribute functions like a name attribute in an HTML <a> tag:

```
<par id="text_and_video">
  <video src="video2.rm" region="newsregion"/>
  <textstream src="text.rt" region="textregion"/>
</par>
```

You then link the source SMIL file to the named target by including a pound sign ("#") and the target name within the link URL. Assuming the target SMIL file is named newmedia.smil, the source file's link to the <par> group looks like this:

```
<a href="rtsp://realserver.example.com/newmedia.smil#text_and_video">
  <video src="video.rm" region="videoregion"/>
</a>
```

Note that the target SMIL file defines two regions, newsregion and textregion. When RealPlayer receives the new SMIL file, it creates those regions as specified in the file's header.

Tips for Linking to a SMIL File

Note the following when linking to another SMIL file:

- You can link to any clip or <par> or <seq> group by defining an id attribute for the clip or group. Do not link to an element in a SMIL file header, however. Although you can link to a <switch> group, do not link to a clip or group within a <switch> group.
- You cannot link to a clip in a <par> group and exclude the other clips in that group. All clips in the group will play in their designated regions.
- If additional clips follow the target clip in the SMIL file, those clips play when the target clip finishes playing.

> **Tip**
>
> If you want to link to a single clip but the SMIL file that refers to that clip lists other clips as well, link to the clip directly. Or create a new SMIL file that lists only the single target clip.

- To link to a target within the same SMIL file, simply set the href attribute value to the target id, such as <a href="#text_and_video">. Be sure to include the pound sign before the id value.

Linking with a Timeline Offset

You can use the <anchor> tag's time coordinates to create a timeline offset in a linked clip. Suppose you want to link a video to another video at 30 seconds into the second video's timeline. In the source SMIL file, you define an <a> or <anchor> link from the first video to a SMIL file that contains the second video. In the second SMIL file, the video's <anchor> tag defines the timeline offset using SMIL timing parameters.

Here is a sample of the link in the first SMIL file:

```
<a href="rtsp://realserver.example.com/newmedia.smil#vid2">
  <video src="video.rm" region="videoregion"/>
</a>
```

The following is the linked video clip in the second SMIL file, newmedia.smil:

```
<video src="video2.rm" region="newsregion">
  <anchor id="vid2" begin="30s"/>
</video>
```

Additional Information

"Specifying Timing" on page 92 describes the SMIL timing values.

Adding Presentation Information

The SMIL file header can use <meta> tags to list presentation information such as title, author, and copyright:

```
<head>
  <meta name="title" content="Bob and Susan Discuss Streaming Media"/>
  <meta name="author" content="RealNetworks Media Productions"/>
  <meta name="copyright" content="(c)1998 RealNetworks"/>
  <meta name="abstract" content="Bob and Susan, two Internet technology
experts, discuss the future of streaming media."/>
 </head>
```

This example defines a title, author, copyright, and abstract. You can define other values as well, such as an e-mail address, simply by adding an attribute such as name="email". This information displays when the user gives the RealPlayer **Help>About this Presentation** command. In addition, the title

displays at the top of the RealPlayer window and in the run list under the RealPlayer **File** menu.

Tip

Name values, as in `name="title"`, must be lowercase. When defining long content such as an abstract, don't use line breaks or tabs within a `content` value.

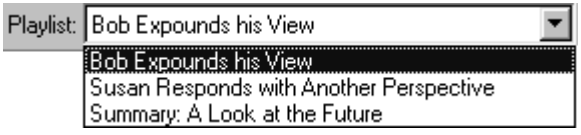
Managing the Playlist

In addition to the presentation information, RealPlayer keeps a playlist of clip titles. Through this playlist, users can play different parts of the SMIL presentation. The following example shows the same header information as shown previously, but adds titles for each video clip in the body section:

```
<smil>
 <head>
  <meta name="title" content="Bob and Susan Discuss Streaming Media"/>
  <meta name="author" content="RealNetworks Media Productions"/>
  <meta name="copyright" content="(c)2000 RealNetworks"/>
  <meta name="abstract" content="Bob and Susan, two Internet technology
experts, discuss the future of streaming media."/>
 </head>
 <body>
  <seq>
   <video src="clip1.rm" title="Bob Expounds his View"/>
   <video src="clip2.rm" title="Susan Responds with Another Perspective"/>
   <video src="clip3.rm" title="Summary: A Look at the Future"/>
  </seq>
 </body>
</smil>
```

The titles for the individual clips appear in RealPlayer's pull-down playlist as shown in the following illustration. The user can also view the playlist with RealPlayer's **View>Playlist** command.

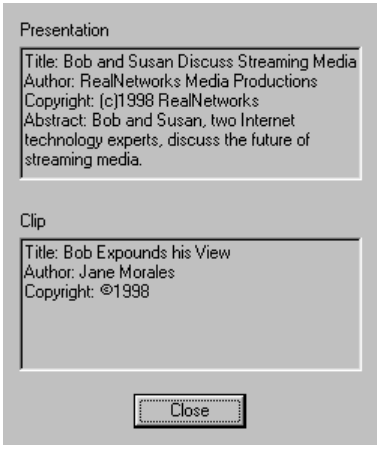
RealPlayer's Pull-Down Playlist Showing Clip Titles



Only the clip titles affect the playlist, but you can also define an author, copyright, and abstract for each clip. As illustrated below, clip information

appears when the RealPlayer user gives the **Help>About this Presentation** command while the clip plays.

Presentation and Clip Information



Note that in this illustration, the clip information section shows the clip title defined in the SMIL file, but also includes author and copyright information not defined through SMIL. The author and copyright information is encoded in the clip.

Tips for Defining Clip Information

The following points explain the relationship between clip information set through SMIL and information encoded in a clip.

- Clips may include their own title, author, copyright, and abstract information. With RealVideo, for example, this information is encoded within the clip. For RealPix, it is set within the RealPix text file.
- When you encode a clip with RealProducer, the information you enter in RealProducer's **Description** field becomes the clip abstract.
- You can override any title, author, copyright, or abstract information encoded in a clip by adding `title`, `author`, `copyright`, and `abstract` attributes to the clip's SMIL source tag:

  ```
  <ref src="..." title="title" author="name" copyright="copyright"
  abstract="abstract"/>
  ```

- To make the playlist useful, always define a title for each clip through SMIL. RealPlayer reads the SMIL titles when the presentation begins. It

reads a clip's encoded title only when the clip starts playback. So if you define titles through SMIL, RealPlayer can immediately build a playlist that allows the user to choose different clips.

- Only `title`, `author`, `copyright`, and `abstract` attributes are supported for clips. For the entire presentation, however, you can define any information through the header section `<meta…/>` tags. See "Adding Presentation Information" on page 118 for more information.
- For a parallel group, use `title`, `author`, `copyright`, and `abstract` attributes in the `<par>` tag instead of the clip source tags. Clip titles are ignored, and only the group title shows in the playlist. For more on `<par>`, see "Playing Clips in Parallel" on page 89.
- A `title`, `author`, `copyright`, or `abstract` attribute in a `<seq>` tag is ignored. This information must be defined for individual clips or parallel groups.
- Although it's easier to set clip information through SMIL, it's good practice to encode the title, author, and copyright information in the clip as well. This preserves the clip information in case you stream the clip without using a SMIL file.

Using Coded Characters

In a header, SMIL interprets quotation marks, apostrophes, ampersands, and angle brackets as syntax markers.To have these characters show up as text in RealPlayer, you use codes in the header. As shown in the following table, codes begin with an ampersand ("&") and end with a semicolon (";"). SMIL interprets these codes the same way as popular Web browsers.

SMIL Coded Characters

Code	Character	Example
`&quot;`	quotation mark	"
`&amp;`	ampersand	&
`&apos;`	apostrophe	'
`&lt;`	left angle bracket ("less than" sign)	<
`&gt;`	right angle bracket ("greater than" sign)	>

For example, to add the following as a title:

```
"Multimedia's <smil> & you"
```

You enter this in the SMIL file header:

```
<meta name="title" content=
 "&quot;Multimedia&apos;s &lt;smil&gt; &amp; you&quot;"/>
```

EXTENDING SMIL

Chapter 6 covers the basics of SMIL, the industry-standard language for streaming multimedia. This chapter describes RealNetworks' SMIL extensions, which enhance SMIL's playback ability. These extensions work only with RealSystem, not with any other SMIL-based media player. Use them only when streaming clips from RealServer to audiences using RealPlayer.

Defining Image Options

For a still image, you can use the options listed in the following table to modify the image's streaming characteristics, or to link it to RealPlayer commands. These options are not SMIL parameters. They are extensions to the image's SMIL source tag, which is described in "Specifying Clip Locations" on page 85. The table summarizes the image options, which are described in detail in the sections that follow.

Options for Image Source Tag

Option	Function	Format	Values	Example
`bitrate`	Sets streaming bit rate.	All	Bits per second	`bitrate=1000`
`bgcolor`	Substitutes color for transparency.	GIF, PNG	Color name or hexadecimal value	`bgcolor=black` `bgcolor=AA3344`
`url`	Links to URL or a RealPlayer command.	All	URL or command	`url=http://www.real.com` `url=command:stop()`
`target`	Sets browser or RealPlayer as target.	All	`_browser` `_player`	`target=_browser` `target=_player`
`reliable`	Ensures image arrival.	All	`true`	`reliable=true`

A question mark operator (?) separates image options from the image URL in the SMIL source tag. To use multiple image options, insert a question mark

before the first option, and separate the remaining options with ampersands (&). The order of options does not matter. The following example shows the basic syntax for adding two image options to the image source tag:

```
<img src="URL?option=value&option=value"/>
```

Note that image options and values are not in quotation marks because they are part of the quoted src value. The following example combines bit rate and background color options:

```
<img src="button.gif?bitrate=1000&bgcolor=blue"/>
```

Setting Image Bit Rates

By default, an image streams at 12 Kbps until RealPlayer receives it.You can set a higher bit rate to take advantage of bandwidth availability and stream an image quickly. Or you can set a lower bit rate when streaming images in parallel with another clip so that the other clip has enough bandwidth for uninterrupted playback. The following example shows three sequential image files set to stream at a low bit rate to ensure that a video playing in parallel does not stall:

```
<par>
 <video src="video.rm" region ="videoregion"/>
 <par>
  <seq>
   <img src="ad1.gif?bitrate=1000" begin="30s" dur="60s" region="adregion"/>
   <img src="ad2.gif?bitrate=1000" dur="120s" region="adregion"/>
   <img src="ad3.gif?bitrate=1000" dur="60s" region="adregion"/>
  </seq>
 </par>
</par>
```

In this example, each GIF image is set to stream at 1 Kbps. This ensures that each image consumes a small amount of bandwidth and does not interfere with video playback.

Note

In the preceding example, the <par> tag just outside the <seq> tag makes RealServer balance the bit rates of the images as a combined group. For more on using <par> to balance bandwidth, see "Smoothing Transitions Between Clips" on page 205.

Keep in mind that the image size divided by the maximum bit rate equals the minimum amount of time needed to stream the image. Assuming that images in the preceding example are 30 kilobits apiece, each image takes about 30 seconds to stream. For this reason, the first image is set to display 30 seconds after the video begins to play. If no begin time were defined, the video would be delayed up to 30 seconds while the first image streams to RealPlayer. As well, each successive image must be able to stream while the preceding image displays.

Overriding GIF or PNG Transparency

For GIF or PNG images that include transparency, you can use `bgcolor` to substitute a color for the transparency. Choose any hexadecimal color value without the leading pound sign ("#"), or any predefined color name as described in "Adding Background Colors" on page 103. If you do not set the color, the region's defined background color shows through transparent spots in the image. Here is an example of an hexadecimal background color defined for an image:

```
<img src="button.gif?bgcolor=BB21AA"/>
```

Linking to a Web Page or Clip

Using the image `url` and `target` options, you can link an image with a fully qualified URL. When the viewer clicks the image, the URL opens in the viewer's browser or RealPlayer. The following example shows an image linked with a URL that opens in the viewer's default browser:

```
<img src="home.gif?url=http://www.example.com&target=_browser" region="home"/>
```

This next example shows an image linked with a URL that opens in RealPlayer:

```
<img src="go.gif?url=rtsp://realserver.example.com/media.smil&target=_player"
  region="next"/>
```

Keep in mind that opening a URL in RealPlayer replaces the current presentation with the new clip or SMIL file.

Note

The image hyperlink options duplicate the SMIL hyperlink features described in "Linking to Other Media" on page 112. You can use either method, depending on your preference.

Controlling RealPlayer

With url=command:*command*&target=_player, you can turn an image into a link that causes RealPlayer to stop, pause, play, or seek to a specific point in the presentation timeline. The following example shows three images set to appear in three separate SMIL regions. Each image issues a different RealPlayer command when clicked:

```
<par>
  <img src="play.gif?url=command:play()&target=_player" region="play"/>
  <img src="pause.gif?url=command:pause()&target=_player" region="pause"/>
  <img src="stop.gif?url=command:stop()&target=_player" region="stop"/>
</par>
```

An image can also issue a seek command that specifies a certain point in the presentation timeline:

```
<img src="seek.gif?url=command:seek(1:35.4)&target=_player"
  region="seek"/>
```

In the preceding example, clicking the image instructs RealPlayer to seek to 1:35.4 in the presentation. This option does not use SMIL shorthand timing values such as s for seconds or min for minutes. Instead, the time format is like the "Normal Play Time Format" described on page 92, except that it also accepts milliseconds and days as values:

```
dd:hh:mm:ss.xyz
```

Here, dd is days, hh is hours, mm is minutes, ss is seconds, x is tenths of seconds, y is hundredths of seconds, and z is milliseconds. Only the ss field is required.

When the time value does not include a decimal point, the last field is read as the seconds. For example, 1:30 means 1 minute and 30 seconds, whereas 1:30:00 means 1 hour and 30 minutes. Thus, all of the following values seek 90 minutes into the presentation timeline:

```
url=command:seek(1:30:00.0)&target=_player
url=command:seek(90:00)&target=_player
url=command:seek(5400)&target=_player
```

Ensuring Reliable Image Transmission

Use the reliable=true option to indicate that an image must be delivered to RealPlayer under any circumstances:

```
<img src="button.gif?reliable=true"/>
```

During extremely adverse network conditions, RealSystem will halt the presentation if necessary rather than drop the image. Use `reliable="true"` sparingly, though, because RealSystem generally ensures that very little data loss occurs in transmission.

Popping Up New RealPlayer Windows

Using RealPlayer 7 or later, you can open as many playback windows as the computer CPU and memory allow. This way, you can keep navigation information visible in one window, for example, while playing a clip in another window. Windows pop up when a viewer clicks a specially configured hyperlink. Instead of opening a Web page or another media clip, though, the hyperlink sends an "open window" command to RealPlayer.

Additional Information

For information on supporting both multiple windows in newer versions of RealPlayer and a single window in RealPlayer G2, see "Switching Presentations for Different RealPlayer Versions" on page 211.

Creating a Link for a New Window

You open a new RealPlayer window by using a hypertext link in a RealText, SMIL, or Flash clip. RealText, SMIL, and the **Get URL** command in Flash all support HTML-style hypertext links. Unlike HTML, though, RealSystem markup tags are case-sensitive. A RealSystem hyperlink that opens a new RealPlayer window uses the following format:

```
<a href="command:openwindow(name, URL, playmode=value, ...)">...</a>
```

When a viewer clicks a link that uses this syntax, `command:openwindow` tells RealPlayer to open a new window for the given URL and stop the presentation in the current window. This command requires two arguments, *name* and *URL*. The *playmode* arguments are optional. You can separate arguments with a comma, but this is not required. A space may precede or follow a comma. If an argument contains characters such as commas or parentheses, enclose it in single quotation marks.

name

The mandatory *name* argument comes first. It supplies a predefined or user-defined name for the new RealPlayer window. The following table lists and describes the values for the *name* argument.

name Argument

Name	Function
`_new` or `_blank`	Opens a new RealPlayer window each time the viewer clicks the link. Each subsequent link named `_new` or `_blank` opens a new window as well.
`_self` or `_current`	Opens the URL in the current RealPlayer window.
name	Creates a new RealPlayer window with the user-defined name. A subsequent `openwindow` command using the same name opens the given URL in the same window.

URL

Following the *name* argument, the required *URL* argument gives the fully qualified URL to the clip or SMIL presentation to play in the new window. You must include the protocol (`rtsp://`, `http://`, `chttp://`, or `file://`) in the URL. Note that relative URLs do not work.

playmode

The optional *playmode=value* argument defines the state of the new RealPlayer window that opens. A command to open a new window can have more than one `playmode` argument. The following table lists and describes the possible `playmode` values.

playmode Attributes and Values

Attribute	Value	Function
`autosize`	`true`	Puts the window in autosize mode so that it is minimized to just the display window when the cursor is not over it.
	`false`	Opens the window in compact mode so that it does not change size on a mouseover. This is the default. Note that normal view mode is available only in the main RealPlayer window, not in pop-up windows.

(Table Page 1 of 2)

playmode Attributes and Values (continued)

Attribute	Value	Function
zoomlevel	normal	Plays clip at its normal encoded size. This is the default.
	double	Doubles clip size.
	full	Plays clip at full-screen. If the operating system does not support full-screen zoom, normal mode is used instead.
ontopwhile playing	true	Keeps window on top of other windows on the desktop.
	false	Lets the user determine which windows to place on top. This is the default.

(Table Page 2 of 2)

Note

You can also open the initial presentation in double, full-screen, or compact mode by using a Ram file. For details on doing this, as well as guidelines for using double and full-screen modes, see "Setting a Presentation's Starting Mode" on page 176.

Examples of Opening New RealPlayer Windows

This section provides examples of hyperlinks that open new RealPlayer windows. The first three sections illustrate how to target various windows with the hyperlink syntax. They use RealText links and Flash **Get URL** commands as examples. The fourth example shows a hyperlinked image in SMIL. The last sample illustrates a SMIL hotspot link for a video clip.

Targeting the Same Window with Multiple Links in RealText or Flash

The following RealText link opens a URL in a new RealPlayer window named `feature`. The new window is set to autosize mode:

```
<a href="command:openwindow(feature, rtsp://realserver.example.com/comedy.rm,
autosize=true)">Comedy Hour</a>
```

In Flash, the **Get URL** command looks like this:

```
command:openwindow(feature, rtsp://realserver.example.com/comedy.rm,
autosize=true)
```

When first clicked, this link creates a RealPlayer window named `feature`. If another link also targets the `feature` window, clicking that link starts the new URL in the `feature` window. Clicking the link in the following example starts

an animal program in the window running the comedy program. Note, however, that this RealText link switches the window out of autosize mode:

```
<a href="command:openwindow(feature, rtsp://realserver.example.com/animals.rm,
autosize=false)">Sharks!</a>
```

The Flash **Get URL** version looks like this:

```
command:openwindow(feature, rtsp://realserver.example.com/animals.rm,
autosize=false)
```

Opening Separate Windows with RealText or Flash

Each link opens a separate window if the window names are different, or you use the predefined name `_new` or `_blank`. The following RealText links open separate autosizing windows that play on top of all other desktop windows:

```
<a href="command:openwindow(_new, rtsp://realserver.example.com/comedy.rm,
autosize=true, ontopwhileplaying=true)">Comedy Hour</a>

<a href="command:openwindow(_blank, rtsp://realserver.example.com/animals.rm,
autosize=true, ontopwhileplaying=true)">Sharks!</a>
```

In Flash, the **Get URL** commands look like this:

```
command:openwindow(_new, rtsp://realserver.example.com/comedy.rm,
autosize=true, ontopwhileplaying=true)

command:openwindow(_blank, rtsp://realserver.example.com/animals.rm,
autosize=true, ontopwhileplaying=true)
```

Launching Clips in the Current Window through RealText or Flash

Use either `_current` or `_self` to open the URL in the current window. The following example is for RealText:

```
<a href="command:openwindow(_self,
rtsp://realserver.example.com/comedy.rm)">Comedy Hour</a>
```

The next RealText link plays the clip at double its encoded size:

```
<a href="command:openwindow(_current,
rtsp://realserver.example.com/animals.rm, zoomlevel=double)">Sharks!</a>
```

The following are the same commands issued through **Get URL** in Flash:

```
command:openwindow(_self, rtsp://realserver.example.com/comedy.rm)

command:openwindow(_current, rtsp://realserver.example.com/animals.rm,
zoomlevel=double)
```

Linking from a SMIL File Image

A SMIL file does not display text on-screen, so it does not support the text links described in the preceding samples. However, SMIL can make hyperlinks from GIF, JPEG, and PNG images, or from any other type of clip that displays on-screen. For example, you can use an image file as a button that opens a new RealPlayer window. The following example shows a hyperlinked image file within a SMIL file:

```
<smil>
 ...
  <a href="command:openwindow(_new, rtsp://realserver.example.com/animals.rm)">
   <img src="launch_button.gif" region="button"/>
  </a>
 ...
</smil>
```

Note that the hyperlink syntax is the same as described for RealText in the preceding sections.

Additional Information

For basic information on SMIL hyperlinks, see "Linking to Other Media" on page 112.

Creating a SMIL Hotspot Link

With SMIL, you can create hotspot links ("image maps") over any clip. The following example defines a hotspot link over the upper-left quadrant of a RealVideo file. Clicking this hotspot activates the hyperlink URL in a new RealPlayer window:

```
<smil>
 ...
  <video src="rtsp://realserver.example.com/video/video1.rm" region="video">
   <anchor href="command:openwindow(_new,
    rtsp://realserver.example.com/video/video2.rm)" coords="0,0,25%,25%"/>
  </video>
 ...
</smil>
```

Additional Information

See "Defining Hot Spot Links" on page 113 for the basics of creating hot spots.

Caching Files on RealPlayer

Using RealPlayer 7 or later, you can cache on disk any files delivered through HTTP. You may want to cache GIF, JPEG, or PNG images that are part of a SMIL presentation, for example. Caching images is beneficial for viewers who reload or revisit presentations frequently. An example is an Internet radio station that uses GIF logos and on-screen buttons. As long as the GIFs reside in the RealPlayer cache, the server does not have to resend the files if, for example, the user clicks a link that opens a new SMIL presentation containing the same images.

Caching works only for files delivered through HTTP. You should not try to cache large clips that would be served better through RTSP, such as video, audio, RealText, Flash, and RealPix clips. (RealPlayer caches RealPix images in memory, but not on disk, for the duration of the RealPix presentation.) Nor should you cache ads or images that do not appear consistently in your presentation.

RealPlayer 7 or later caches files within the RealPlayer home directory in a folder named `cache_db`. Its cache is independent of any Web browser cache. The default RealPlayer cache size is 4 MB, although the user can change the size. The cache is limited in size because it is meant to hold small images such as buttons and logos, not large graphics or streaming clips.

Note

Caching does not work with RealPlayer G2. For information on using caching with later versions of RealPlayer while still supporting RealPlayer G2, see "Switching Presentations for Different RealPlayer Versions" on page 211.

Using the CHTTP Caching Protocol

RealPlayer does not cache all items streamed by HTTP. Instead, you designate files to cache by using `chttp://` instead of `http://` in the file URLs. When RealPlayer reads a CHTTP URL in a SMIL file, it first checks its disk cache for the file. If the file is not present, RealPlayer requests the file through HTTP, storing the file in its cache. Because RealPlayer interprets a `chttp://` URL as a special instance of HTTP downloading, caching works for any file stored on an HTTP-compatible server.

If a file is stored in RealPlayer's cache, RealPlayer reuses the file instead of requesting it again from the server as long as a CHTTP URL is used. The cached version is not used, though, if the URL starts with `http://` or differs in any way from the original CHTTP URL. Also, any new file requested through RTSP, HTTP, or PNA is not cached. The following SMIL example indicates that the specified GIF image should be downloaded and cached for later use:

```
<img src="chttp://www.example.com/images/image1.gif" region="image"/>
```

This URL causes RealPlayer to store `image1.gif` in the cache for later use until the cached file expires or is overwritten. For the cache expiration rules, see "Cache Size and Expiration Rules" on page 133.

Controlling the RealPlayer Cache

Because RealPlayer supports the same HTTP header fields used to control file expiration in Web browser caches, it can carry out caching directives set by Web servers. Thus, you can reuse Web page images in RealPlayer presentations without losing control of how these images are cached. This section describes how to use HTTP headers to control the RealPlayer cache, and how RealPlayer manages its cache. Documentation for most Web servers includes information about how to set fields in HTTP header files.

Overriding Caching with Cache-Control

The `Cache-Control` command of an HTTP header file can override caching of a RealPlayer file requested through `chttp://`. A file requested through CHTTP is not cached if any of the following are present as meta-information in the HTTP header file:

- `Cache-Control:no-cache`
- `Cache-Control:no-store`
- `Cache-Control:private`
- `Cache-Control:must-revalidate`

Cache Size and Expiration Rules

By default, RealPlayer's HTTP cache size is set to 4 MB, although users can modify this size. Unless an HTTP header sets a file lifetime, the cached file expires after 4 hours. A subsequent request for a cached item restarts the item's expiration clock. As the cache fills, RealPlayer begins to delete unexpired items to reclaim needed disk space on a first-in, first-out basis.

Changing the Lifetime of a Cached File

Within an HTTP header, you can have `Cache-Control:max-age` set the "time to live" (TTL) for a cached file, overriding the default expiration time. Expressed in seconds, the maximum age is added to the current time to yield the file's expiration time. This value must be between 60 seconds and one year. For example:

```
Cache-Control:max-age=172800
```

If you do not use the `Cache-Control:max-age` field, you can have the `Expires` field determine the file's expiration time. The `Expires` field takes as an attribute a date string that defines when the cached element expires, relative to the caching computer's clock. The date string is formatted as follows:

`Expires=` *`Wdy, DD Mon YYYY HH:MM:SS`* `GMT`

The weekday is optional. In the following two examples, the first example includes a weekday designation, the second one does not:

```
Expires= Fri, 17 Mar 2000 19:37:09 GMT
```

```
Expires= 17 Mar 2000 19:37:09 GMT
```

The weekday and month abbreviations are as follows:

Day of week:	`Mon, Tue, Wed, Thu, Fri, Sat, Sun`
Month:	`Jan, Feb, Mar, Apr, May, Jun, Jul, Aug, Sep, Oct, Nov, Dec`

Note

The entry is not cached if the value in the `Expires:` field predates the current date and time.

User Control of the RealPlayer Cache

RealPlayer users can control some aspects of RealPlayer's cache by disabling the cache, setting the amount of disk space available for the cache, and emptying the cache. Users carry out these actions through the RealPlayer preferences. For more information, see the RealPlayer online help.

Authoring SMIL Files for Caching

To cache files, write your SMIL file to download the cached items before other streamed elements. You can do this by placing the cached elements in a SMIL `<seq>` group ahead of the streamed elements. The following SMIL extract, which omits the header information for convenience, caches two logo images.

The SMIL file is for a presentation in which logos stay the same but the streaming RealVideo and RealText clips change frequently:

```
<smil>
  ...header omitted from example...
  <body>
    <seq>
      <!-- First, download and cache these two logos. -->
      <img src="chttp://www.example.com/images/logo1.gif?bitrate=20000"
        region="logo1" fill="freeze"/>
      <img src="chttp://www.example.com/images/logo2.gif?bitrate=20000"
        region="logo2" fill="freeze"/>
      <par>
        <!--Second, stream these 2 clips in parallel. -->
        <textstream src="rtsp://realserver.example.com/news.rt"
          region="newsregion" fill="freeze"/>
        <video src="rtsp://realserver.example.com/newsvid.rm"
          region="videoregion"/>
      </par>
    </seq>
  </body>
</smil>
```

In this example, the two logos quickly download in sequence. Next, RealServer streams a parallel group comprising the RealVideo and RealText clips. When RealPlayer plays this presentation again, it first checks its cache for the two logos. If it finds them, it skips directly to the streaming clips.

PLAYING CLIPS IN A WEB PAGE

Embedding a presentation in a Web page lets you play back clips without launching RealPlayer as a separate application. You can even include in your page RealPlayer controls for actions such as fast-forward and pause. This chapter explains how you add markup to a Web page so that people view your RealSystem presentation directly through their Web browsers.

Additional Information

For a quick example of how to embed a video in a Web page, see "Embedding a RealVideo Clip in a Web Page" on page 201.

Choosing the Netscape Plug-in or ActiveX Control

To provide Web page playback, RealPlayer includes a plug-in for browsers that support the Netscape plug-in architecture:

- Netscape Navigator 3.0 and later.
- Microsoft Internet Explorer 3.0 and later.

RealPlayer also has an ActiveX control that provides playback capabilities within the following products:

- Microsoft Internet Explorer 3.0 and later.
- Any application that supports ActiveX controls, such as Microsoft Visual Basic, Microsoft Visual C++, Microsoft Access, and so on.

Because they both have the same capabilities, you can use either the plug-in or the ActiveX control, depending on which products you need to support. The following sections describe the basics of using the plug-in or the control and explain each parameter you can set.

Note

Familiarity with RealPlayer and HTML will make it easier to use the plug-in or the control. In the following examples, ellipses (...) indicate information left out to keep the examples short. Make sure that your tags include all necessary information.

Additional Information

Embedded RealPlayer Extended Functionality Guide at **http://service.real.com/help/library/encoders.html** explains how to use JavaScript or VBScript to extend the functionality of the Netscape plug-in or the ActiveX control.

Using <EMBED> Tags for the Netscape Plug-In

To use RealPlayer's Netscape plug-in, you add `<EMBED>` tags to your Web page HTML. Each `<EMBED>` tag has three required parameters (`SRC`, `WIDTH`, `HEIGHT`) and can include many optional parameters. The basic `<EMBED>` tag looks like the following (the `SRC` value, which is explained below, has been omitted for simplicity):

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134>
```

This tag creates a playback area 300 pixels wide by 134 pixels high within the Web page. Parameters are typically in the form `PARAMETER=value`. Parameter names can be any case, although this manual shows them uppercase. Except for file names, which typically must be lowercase, parameter values are not case-sensitive. Unless they are URLs, parameter values do not need to be enclosed in quotation marks.

Supporting Other Browsers

To accommodate browsers that do not support the Netscape plug-in, use `<NOEMBED>` to define a standard hypertext link to your presentation. The unembedded link follows the `<EMBED>` tag:

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134>
<NOEMBED><A HREF="...">Play with RealPlayer.</A></NOEMBED>
```

In this example, browsers that can play the embedded presentation hide the text between `<NOEMBED>` and `</NOEMBED>`. Other browsers ignore the preceding

<EMBED> tag and display only the hypertext link. The user then clicks the link to play the presentation in RealPlayer.

Using <OBJECT> Tags for the ActiveX Control

You embed the RealPlayer ActiveX control in HTML pages by using the <OBJECT> tag. This tag uses an ID that you select, such as ID=RVOCX, and it must have the following class ID, which identifies RealPlayer:

```
CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
```

The <OBJECT> tag also sets the width and height of the playback area within the browser. A typical <OBJECT> tag looks like this:

```
<OBJECT ID=RVOCX CLASSID="clsid:CFCDAA03-8BE4-11cf-B84B-0020AFBBCCFA"
WIDTH=300 HEIGHT=134>
... parameters ...
</OBJECT>
```

This tag creates a playback area 300 pixels wide by 134 pixels high within the Web page. Between <OBJECT> and </OBJECT>, you can define any number of additional parameters with the following syntax:

```
<PARAM NAME="name" VALUE="value">
```

PARAM, NAME, and VALUE markers can be any case, although in this book they are uppercase. Parameter values are not case-sensitive, except for file names, which typically must be lowercase. Always enclose parameter values in double quotation marks.

Setting Basic Parameters

Both the Netscape plug-in and the ActiveX control use the same basic tag parameters. As explained in the preceding section, however, the tag syntaxes for the plug-in and the control differ. This section explains the basic parameters you can include in each <EMBED> or <OBJECT> tag.

SRC

The SRC parameter gives the presentation's source URL enclosed in double quotation marks. The directory names cannot contain spaces. For the ActiveX control, the <OBJECT> tag's CLASSID parameter causes the presentation to play in the Web page, so you can simply link to the SMIL file or clip within one <OBJECT> tag. When linking to a presentation on RealServer, you can include

the Ramgen parameter in the URL, as in the following example (although this is not necessary):

```
<PARAM NAME="SRC" VALUE="http://realserver.example.com:8080/ramgen/sample.smil">
```

When you use the Netscape plug-in, you need to include the SRC parameter in every `<EMBED>` tag. You can specify two types of URLs. The first uses the `?embed` option of the Ramgen parameter, which causes RealPlayer to play the clip or SMIL presentation through its Netscape plug-in instead of launching as a separate application:

```
SRC="http://realserver.example.com:8080/ramgen/sample.smil?embed"
```

Additional Information

For more on Ramgen and presentation URLs, see "Linking Your Web Page to RealServer" on page 168.

The second option is to create a Ram file with a `.rpm` extension and list this file in the `SRC` parameter without including Ramgen and the `?embed` option. The Ram file then links to your SMIL file or clips. The following example links to a `.rpm` file on a Web server:

```
SRC="http://www.example.com/media/sample.rpm"
```

Additional Information

For instructions on writing a `.rpm` file, see "Creating a Ram File Manually" on page 173.

You must create and link to a `.rpm` file in any of the following situations:

- Your RealServer does not use Ramgen.
- The SMIL file or clip is on a Web server.
- All files are located on a user's local computer rather than streamed over a network.

Note

When linking to local files, you can use relative links just as you do in HTML. You can also use absolute links. When using absolute links with the Netscape plug-in, use forward slashes in paths. For example, instead of `file://c:\media\first.rpm`, set the source as `file://c:/media/first.rpm`.

WIDTH and HEIGHT

Required for each <EMBED> or <OBJECT> tag, the WIDTH and HEIGHT parameters set the size of the playback area. If you omit these parameters, the playback area may appear as a tiny icon because streaming media presentations do not resize themselves automatically. The values for WIDTH and HEIGHT are in pixels by default, so a width of 300 creates a playback area 300 pixels wide. Setting WIDTH and HEIGHT to 0 (zero) hides the playback area.

You can also express WIDTH and HEIGHT as percentages of the browser window size. For example, a width of 50% makes the width of the presentation area half the width of the browser window. Keep in mind that different types of media scale with different results. For example, a video scaled larger than its encoded size may not look good. Vector-based clips such as Flash animations, on the other hand, scale more easily to fit different playback areas.

NOJAVA

Setting the NOJAVA parameter to true in every <EMBED> tag prevents the browser's Java Virtual Machine (JVM) from starting if it is not yet running:

```
<EMBED SRC="..." WIDTH=300 HEIGHT=134 NOJAVA=true>
```

This is recommended when using the Netscape plug-in because the JVM is not required for the <EMBED> parameters described in this chapter. Starting the JVM therefore delays presentation playback unnecessarily. The JVM is required only when you are extending plug-in functionality with JavaScript. In this case, omit NOJAVA entirely from the <EMBED> tags.

Additional Information

See *Embedded RealPlayer Extended Functionality Guide* at **http://service.real.com/help/library/index.html** for more information on controlling RealPlayer with JavaScript.

Note

Although you can include NOJAVA as an ActiveX parameter, it has no effect on Internet Explorer, which starts its JVM when the browser is launched.

Adding RealPlayer Controls

With the CONTROLS parameter, you can add RealPlayer controls such as a play/pause button to your Web page. Visitors to your page can then control the presentation playback as if they were using RealPlayer as a separate application. The following example for the Netscape <EMBED> tag displays the play/pause button in your Web page:

```
<EMBED SRC="..." WIDTH=26 HEIGHT=26 NOJAVA=true CONTROLS=PlayButton>
```

For the ActiveX control, you define a CONTROLS parameter within the <OBJECT> tag structure:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=26 HEIGHT=26>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
</OBJECT>
```

The following sections describe the embedded RealPlayer controls. You use a tag's WIDTH and HEIGHT parameters to set the control's size. The suggested pixel widths and heights given with the control descriptions produce controls approximately the same sizes as the RealPlayer controls. Specifying different pixel sizes scales the controls larger or smaller. You can also use percentage values for sizes, but this is recommended only for the image window.

Additional Information

For information on adding more than one control to your Web page, see "Linking Multiple Controls" on page 148.

Tip

Download the HTML version of this manual from **http://service.real.com/help/library/encoders.html** to see online examples of embedded RealPlayer controls.

Basic Controls

ImageWindow

Displays the image window. This is available only for display presentations such as video or animation. Even if no other controls are visible on the page, the user can typically right-click (on Windows) or hold down the mouse button (on the Macintosh) in the playback area to display a menu of choices such as **Play** and **Stop**. See also "Controlling Image Display" on page 150.

Suggested pixel width: 176 or greater
Suggested pixel height: 132 or greater

All

Displays the basic RealPlayer control panel. The control name "default" also works. Functions include play, pause, stop, fast-forward, and rewind. Sliders include a position slider and a volume slider with a mute button that pops up when the speaker button is clicked. Below the buttons are a clip information field, a status panel, a network congestion indicator, and a clip timing field.

Suggested pixel width: 375
Suggested pixel height: 100

Individual Controls and Sliders

ControlPanel

Displays a compact RealPlayer control panel. Functions include play, pause, stop, fast-forward and rewind. There's also a position slider, along with a volume slider and mute button that pops up when the speaker button is clicked.

Suggested pixel width: 350
Suggested pixel height: 36

PlayButton

Displays a play/pause button.

Suggested pixel width: 44
Suggested pixel height: 26

PlayOnlyButton

Displays a play button.

Suggested pixel width: 26
Suggested pixel height: 26

PauseButton

Displays a pause button.

Suggested pixel width: 26
Suggested pixel height: 26

StopButton

Displays a stop button.

Suggested pixel width: 26
Suggested pixel height: 26

FFCtrl

Displays a fast-forward button.

Suggested pixel width: 26

Suggested pixel height: 26

RWCtrl

Displays a rewind button.

Suggested pixel width: 26

Suggested pixel height: 26

MuteCtrl

Displays a mute button.

Suggested pixel width: 26

Suggested pixel height: 26

MuteVolume

Displays a mute button and volume slider.

Suggested pixel width: 26

Suggested pixel height: 88

VolumeSlider

Displays a volume slider.

Suggested pixel width: 26

Suggested pixel height: 65

PositionSlider

Displays a clip position slider.

Suggested pixel width: 120

Suggested pixel height: 26

TACCtrl

Displays a clip information field. For instructions on formatting the clip playlist, see "Managing the Playlist" on page 119.

Suggested pixel width: 370

Suggested pixel height: 32

HomeCtrl

Displays the Real™ logo.

Suggested pixel width: 45

Suggested pixel height: 25

Information Panels

InfoVolumePanel

Displays presentation information along with the volume slider and mute button. For more on presentation information, see "Adding Presentation Information" on page 118. See also "NOLABELS" on page 152.

Suggested pixel width: 325

Suggested pixel height: 55

InfoPanel

Displays the presentation information panel. For more on presentation information, see "Adding Presentation Information" on page 118. See also "NOLABELS" on page 152.

Suggested pixel width: 300
Suggested pixel height: 55

Status Panels

StatusBar

Displays the status panel, which shows informational messages. It also includes the network congestion LED and the position field, which shows the clip's current place in the presentation timeline along with the total clip length.

Suggested pixel width: 300
Suggested pixel height: 30

Note

The status bar is included in the `All` control. If you do not embed a status bar or status field in your page, error messages display in the browser's status bar.

StatusField

Displays the message text area of the status bar. If you do not embed a status field or status bar in your page, error messages display in the browser's status bar.

Suggested pixel width: 200
Suggested pixel height: 30

PositionField

Displays the position field, which shows the clip's current place in the presentation timeline and the total clip length.

Suggested pixel width: 90

Suggested pixel height: 30

Linking Multiple Controls

The CONSOLE parameter defines a name that unifies <EMBED> or <OBJECT> tags so that multiple controls work together. For example, you could create three separate <EMBED> or <OBJECT> tags to define an image window, a play button, and a stop button. By using three tags, you can set the size of each control separately and define its layout with HTML tags. For example, you could put each control in a different HTML table cell.

To tie controls together, define the same CONSOLE name within each <EMBED> or <OBJECT> tag, or use one of these predefined names:

_master links the control to all other embedded controls on the page.

_unique links the control to no other embedded controls on the page.

You can have multiple console names for separate presentations. For a page showing two video clips, for example, you can define the console names video1 and video2. All controls linked by video1 interoperate, as do all controls linked by video2. But a video1 volume slider, for example, will not affect the volume of a video2 clip.

Tips for Using Consoles

Note the following when grouping multiple controls with CONSOLE attributes:

- Every <EMBED> tag must have a SRC attribute. Tags linked by a console name should have the same SRC value.
- With the ActiveX control, only one <OBJECT> tag in a console group needs to have a SRC value.
- If the <EMBED> or <OBJECT> tags in a console group have different SRC values, the first valid source that RealPlayer finds among those choices becomes the console source. This may not always be the first source listed.
- Clicking a play button for one console stops playback for other consoles. This allows multiple consoles to play separate audio tracks or to use the same image window.

Multiple Controls Example

The following examples for the <EMBED> and <OBJECT> tags set up an image window and two sets of controls (a play button and stop button) for two separate videos, sample1.rm and sample2.rm. The predefined console name _master links the image window to both control sets. The control sets use different console names, however, so they do not link to each other. Clicking each play button therefore starts a different video.

Netscape Plug-in Sample Markup

Because each <EMBED> tag must have a SRC value, the image window in the following example uses the same source as the first play button. The viewer simply clicks either play button to start a video. Clicking the other play button stops the first video and plays the second one:

```
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample1.rm?embed"
WIDTH=176 HEIGHT=128 NOJAVA=true CONTROLS=ImageWindow CONSOLE=_master>

<H4>Video 1</H4>
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample1.rm?embed"
WIDTH=44 HEIGHT=26 NOJAVA=true CONTROLS=PlayButton CONSOLE=video1>
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample1.rm?embed"
WIDTH=26 HEIGHT=26 NOJAVA=true CONTROLS=StopButton CONSOLE=video1>

<H4>Video 2</H4>
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample2.rm?embed"
WIDTH=44 HEIGHT=26 NOJAVA=true CONTROLS=PlayButton CONSOLE=video2>
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample2.rm?embed"
WIDTH=26 HEIGHT=26 NOJAVA=true CONTROLS=StopButton CONSOLE=video2>
```

ActiveX Control Sample Markup

In the following ActiveX example, only the controls for the two play buttons define the source URLs (for convenience the CLASSID value is omitted):

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=176 HEIGHT=128>
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="CONSOLE" VALUE="_master">
</OBJECT>

<H4>Video 1</H4>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=44 HEIGHT=26>
<PARAM NAME="SRC"
  VALUE="http://realserver.example.com:8080/ramgen/sample1.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<PARAM NAME="CONSOLE" VALUE="video1">
</OBJECT>
```

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=26 HEIGHT=26>
<PARAM NAME="CONTROLS" VALUE="StopButton">
<PARAM NAME="CONSOLE" VALUE="video1">
</OBJECT>

<H4>Video 2</H4>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=44 HEIGHT=26>
<PARAM NAME="SRC"
 VALUE="http://realserver.example.com:8080/ramgen/sample2.rm">
<PARAM NAME="CONTROLS" VALUE="PlayButton">
<PARAM NAME="CONSOLE" VALUE="video2">
</OBJECT>
<OBJECT ID=RVOCX CLASSID="..." WIDTH=26 HEIGHT=26>
<PARAM NAME="CONTROLS" VALUE="StopButton">
<PARAM NAME="CONSOLE" VALUE="video2">
</OBJECT>
```

Controlling Image Display

The following <EMBED> and <OBJECT> parameters control aspects of how clips play back. This example shows two of these parameters used with the Netscape plug-in:

```
<EMBED SRC="..." WIDTH=50% HEIGHT=50% NOJAVA=true
BACKGROUNDCOLOR=gray CENTER=true>
```

This example is for the ActiveX control:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50% HEIGHT=50%>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="BACKGROUNDCOLOR" VALUE="gray">
<PARAM NAME="CENTER" VALUE="true">
</OBJECT>
```

BACKGROUNDCOLOR

This parameter specifies a background color for the image window. The specified background color also shows through if a clip includes transparency. The background color is black by default. You can use an RGB hexadecimal color value (#RRGGBB) or the following color names, shown here with their corresponding RGB values:

white (#FFFFFF)	silver (#C0C0C0)	gray (#808080)	black (#000000)
yellow (#FFFF00)	fuchsia (#FF00FF)	red (#FF0000)	maroon (#800000)
lime (#00FF00)	olive (#808000)	green (#008000)	purple (#800080)

`aqua (#00FFFF)` `teal (#008080)` `blue (#0000FF)` `navy (#000080)`

Note

SMIL region background colors override this background color. For more on setting SMIL region colors, see "Adding Background Colors" on page 103.

CENTER

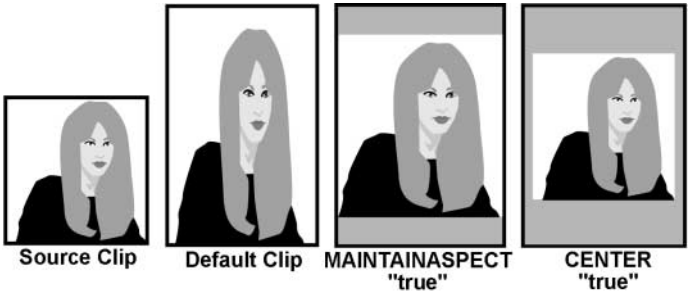
The default value for `CENTER` is false, which causes the clip to fill the entire playback area. If you set `CENTER` to true, the clip is centered within the playback area and is displayed at its encoded size. So by setting `CENTER` to true, you can create a large playback area with `WIDTH` and `HEIGHT` and still have the clip play at its normal size. You cannot use `CENTER` along with `MAINTAINASPECT`.

MAINTAINASPECT

This parameter, which you cannot combine with `CENTER`, determines whether the clip's height-to-width ratio stays constant when the clip scales to fit the image window. The default value of false causes this ratio to change as necessary to fill the image window fully. This may distort the source image.

If you set `MAINTAINASPECT` to true, a clip's height-to-width ratio stays constant. For example, a clip's height-to-width ratio of 1:1 stays constant even if the image window's height-to-width ratio is 2:3. In these cases, the clip is centered in the image window and scaled until one dimension reaches the window's boundaries and the other dimension is within the boundaries. The following illustration shows how clips scale by default, with `MAINTAINASPECT` set to true, and with `CENTER` set to true.

Clip Scaling with MAINTAINASPECT and CENTER



NOLABELS

When you use a control that includes presentation information such as title, author, and copyright, you can include the NOLABELS option to suppress that information in RealPlayer 5. This option does not affect RealPlayer G2 or later.

Additional Information

The "Parameter Reference" section on page 155 lists the options that work with RealPlayer 5.

NOLOGO

When set to true, NOLOGO prevents the Real logo from displaying in the image window. When there are no clips playing, only the specified background color shows in the window. The parameter is false by default.

Setting Automatic Playback

The AUTOSTART, LOOP, and NUMLOOP parameters let you set a presentation to start playing automatically and to loop continuously or for a specified number of times. The following example shows two of these parameters used with the Netscape plug-in:

```
<EMBED SRC="..." WIDTH=50% HEIGHT=50% NOJAVA=true AUTOSTART=true
LOOP=true>
```

This example is for the ActiveX control:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=50% HEIGHT=50%>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="AUTOSTART" VALUE="true">
<PARAM NAME="LOOP" VALUE="true">
</OBJECT>
```

AUTOSTART

When set to true, the AUTOSTART parameter starts playback immediately. When you have multiple <EMBED> or <OBJECT> tags linked by a CONSOLE name, set AUTOSTART to true in just one tag. Leaving AUTOSTART out or setting its value to false means that the presentation will not start until the user starts it by clicking an embedded play button, for example.

LOOP

If the LOOP parameter is set to true, the presentation continuously loops until the viewer stops it. When you have multiple <EMBED> or <OBJECT> tags linked by a CONSOLE name, set looping in just one tag. If you leave the LOOP parameter out, the default value of false applies and the presentation stops after the first playback. The user can play the presentation again by clicking the play button.

NUMLOOP

If you specify a NUMLOOP value such as "2", the presentation loops the specified number of times and then stops. If you use both LOOP and NUMLOOP, the LOOP parameter is ignored.

SHUFFLE

The SHUFFLE parameter is for use only with Ram or SMIL files that list a single sequence of clips. When set to true, SHUFFLE causes RealPlayer to play back the clips in a random order.

Laying Out SMIL Presentations

As explained in "Laying Out Multiple Clips" on page 99, you can use a SMIL file to define separate playback regions for different parts of a presentation. This lets you lay out two clips side-by-side, for example. When playing a presentation in a Web page, you can define the layout in SMIL or in HTML.

Defining the Layout with SMIL

To control the layout by using SMIL, you set up the regions and their relative placements in the SMIL file. You then use the Netscape plug-in or the ActiveX control to create a Web page playback area large enough to accommodate all SMIL regions. This SMIL file then produces the same layout when played through the Web page or RealPlayer.

The sample layout shown in "SMIL Layout Example" on page 108 defines three regions, creating a total playback area 430 pixels wide by 165 pixels high. To accommodate this in your Web page, you define an image window at least as large as this by using either the <EMBED> or the <OBJECT> tag. The following example is for the Netscape plug-in:

```
<EMBED SRC="..." WIDTH=430 HEIGHT=165 NOJAVA=true CONTROLS=ImageWindow
CONSOLE=one>
```

The next example is for the ActiveX control:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=430 HEIGHT=165>
<PARAM NAME="SRC" VALUE="...">
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="CONSOLE" VALUE="one">
</OBJECT>
```

The SRC parameter provides the URL to the SMIL file. You can then use additional <EMBED> or <OBJECT> tags linked to the console named "one" to provide RealPlayer controls for the presentation.

Defining the Layout with HTML

The second method omits SMIL layout information and instead defines the layout by using HTML. You could place an image window and separate RealPlayer controls in an HTML table, for example. Each <EMBED> or <OBJECT> tag then uses a REGION parameter to define a region name. The region each clip plays in is denoted by each clip's source tag within the SMIL file. For example, the SMIL file shown in "SMIL Layout Example" on page 108 lists the following RealText file, which is set to play in the newsregion region:

```
<textstream src="news.rt" region="newsregion"/>
```

Within the HTML page, the <EMBED> tag that plays news.rt would look like this:

```
<EMBED SRC="http://realserver.example.com:8080/ramgen/sample.smil?embed"
WIDTH=250 HEIGHT=144 NOJAVA=true CONTROLS=ImageWindow
REGION=newsregion CONSOLE=one>
```

The <OBJECT> tag would look like this:

```
<OBJECT ID=RVOCX CLASSID="..." WIDTH=250 HEIGHT=144>
<PARAM NAME="SRC"
  VALUE="http://realserver.example.com:8080/ramgen/sample.smil">
<PARAM NAME="CONTROLS" VALUE="ImageWindow">
<PARAM NAME="REGION" VALUE="newsregion">
<PARAM NAME="CONSOLE" VALUE="one">
</OBJECT>
```

You define similar <EMBED> or <OBJECT> tags to create other regions for other clips listed in the SMIL file. The SRC parameter in each tag lists the same SMIL file. You can also use additional <EMBED> or <OBJECT> tags linked to the same console to provide RealPlayer controls for the presentation.

Omitting SMIL Layout Information

When you define the presentation layout through HTML, you must omit the <layout> section from the SMIL file header. For example, the SMIL file shown in "SMIL Layout Example" on page 108 would not have a <layout> section:

```
<smil>
 <head>
  <!--presentation with 2 text clips and 1 video clip-->
  <meta name="title" content="Music of the Week"/>
 </head>
 <body>
  <par>
   <!-- play these 3 clips simultaneously -->
   <textstream src="news.rt" region="newsregion"/>
   <video src="newsvid.rm" region="videoregion"/>
   <textstream src="stocks.rt" region="stockregion"/>
  </par>
 </body>
</smil>
```

Tip

A SMIL file without a layout still works with RealPlayer, but RealPlayer automatically creates the layout, and the results may not be what you expect. Always include a SMIL layout when playing a presentation directly in RealPlayer.

Note

Because there is no SMIL layout, hot spot links are ignored. Note, however, that you can make media clips hyperlinks by using the SMIL <a>… tags. For more on SMIL hyperlinks, see "Linking to Other Media" on page 112.

Parameter Reference

The following table lists the <EMBED> and <OBJECT> tag parameters. Parameters compatible with RealPlayer 5 are indicated in the "5?" column. To have an embedded presentation play back with RealPlayer 5 as well as RealPlayer G2 or

later, use only RealPlayer 5-compatible parameters. Also be sure to use `pnm://` instead of `rtsp://` in the URL.

<EMBED> and <OBJECT> Tag Parameters

Parameter	Function	5?	Values	Default	Reference
AUTOSTART	Sets automatic playback.	**yes**	true|false	false	page 152
BACKGROUNDCOLOR	Sets background color.	no	Color name or RGB hex value	black	page 150
CENTER	Centers clip in the window.	no	true|false	false	page 151
CONSOLE	Links multiple controls.	**yes**	Name, _master, or _unique	(none)	page 148
CONTROLS	Adds RealPlayer controls.	**yes**	Control name	All	page 142
HEIGHT	Sets window or control height.	**yes**	Percentage or pixels	(none)	page 141
LOOP	Loops clips indefinitely.	no	true|false	false	page 153
MAINTAINASPECT	Preserves image aspect ratio.	no	true|false	false	page 151
NOJAVA	Prevents JVM start-up.	no	true|false	false	page 141
NOLABELS	Suppresses presentation information in RealPlayer 5.0.	**yes**	true|false	false	page 152
NOLOGO	Suppresses RealLogo.	no	true|false	false	page 152
NUMLOOP	Loops clip a given number of times.	no	Any number	(none)	page 153
REGION	Ties clip to SMIL region.	no	SMIL region	(none)	page 154
SHUFFLE	Randomizes playback.	no	true|false	false	page 153
SRC	Specifies source clip.	**yes**	URL	(none)	page 139
WIDTH	Sets window or control width.	**yes**	Percentage or pixels	(none)	page 141

INSERTING ADS INTO A PRESENTATION

RealSystem's advertising extension and SMIL make it easy to insert advertisements into your streaming presentation. You can include GIF ad banners, for example, or have a video ad precede each requested clip. This chapter explains how to place ads in SMIL presentations. Before you start, you should understand SMIL, as described in Chapter 6.

How Ad Streaming Works

To use RealSystem's advertising extension, you create a SMIL file that lays out requested clips and ad clips. Instead of listing ad clip URLs in SMIL tags such as `<img/>` or `<video/>`, you write `<RealAdInsert/>` tags that cause RealServer to insert ad URLs automatically. RealServer typically gets these ad URLs from a third-party ad server. RealSystem can deliver ads in any format played by RealPlayer, including GIF, animated GIF, JPEG, PNG, RealAudio, RealVideo, and Flash.

The URL used to request the SMIL file—not the `<RealAdInsert/>` tag—determines what type of ad appears in place of each `<RealAdInsert/>` tag. For example, your Web page might contain a hyperlink to a SMIL file that looks like the following:

```
<a href="http://realserver.example.com/ramgen/adtag/banner/start.smil">...</a>
```

Here, the URL component `/adtag/banner/` indicates that the SMIL file contains one or more `<RealAdInsert/>` tags that RealServer replaces with ad URLs. The RealServer administrator might define `/adtag/banner/` to mean that each `<RealAdInsert/>` tag is replaced by a single 486-pixel-by-60-pixel banner ad. A URL component such as `/adtag/rotating_banner/` might indicate banner ads that rotate every 30 seconds. Another component such as `/adtag/lead_in/` might be for RealVideo ads that precede the requested content.

To use RealSystem's advertising extension, you need to work closely with your RealServer administrator. The administrator can tell you the sizes and file formats of ads available for streaming, the URLs you will need to use in your Web pages, and other important information. RealServer can also generate SMIL files containing ads, or insert ads in existing SMIL presentations. This lets you serve ads for existing content without having to write or modify SMIL files.

Tip

Check with your RealServer administrator on the availability of the SMIL generation feature for ad streaming. This feature can save you time and effort when you want to include ads with existing clips or SMIL files.

Displaying Banner Ads

RealServer can stream a single banner ad or a series of rotating banner ads within a SMIL presentation. Each ad includes a hyperlink that, when clicked, opens the URL in the viewer's Web browser. The RealServer administrator defines the banner ad size and determines how much bandwidth each ad uses. For rotating banner ads, the administrator also determines how frequently RealPlayer displays a new ad.

Note

Before creating SMIL files with banner ads, check with the administrator on banner ad size, bandwidth, type (single ad or rotating ads), and rotation frequency.

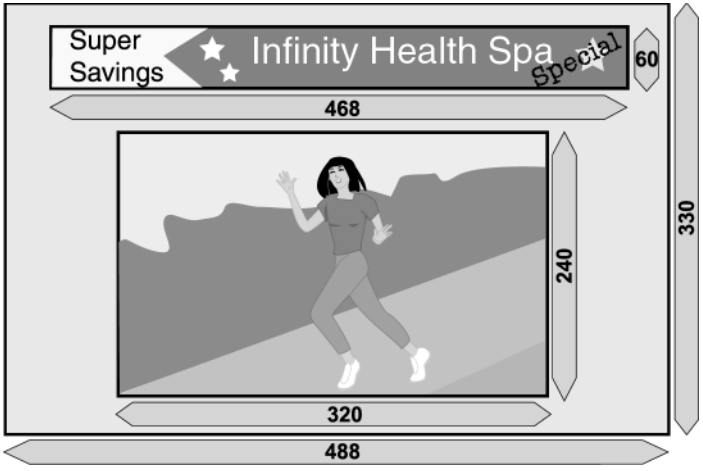
Laying Out the Banner Region

To integrate banner ads with your presentation, you create a SMIL region in which the ad image or rotating image series is displayed. The SMIL file in the following example creates an ad banner region above a video region. The video region is 320 pixels wide by 240 pixels high. The ad banner region is 468 pixels wide by 60 pixels high, which is a common size for ad banners. In the SMIL body section, a `<par>` group plays the requested video and banner ads in their respective regions:

```
<smil>
  <head>
    <!--presentation with video clip and ad banner-->
    <layout>
      <root-layout width="488" height="330" background-color="black"/>
      <region id="ad_banner" top="10" left="10" width="468" height="60"/>
      <region id="videoregion" top="80" left="84" width="320" height="240"/>
    </layout>
  </head>
  <body>
    <par>
      <RealAdInsert region="ad_banner" dur="9:00.0" fill="freeze"/>
      <video src="rtsp://realserver.example.com/videos/newsvid.rm"
        region="videoregion"/>
    </par>
  </body>
</smil>
```

The following illustration shows this layout of a banner ad region above a requested video clip.

Ad Banner Layout



Additional Information

"Laying Out Multiple Clips" on page 99 explains how to set up SMIL regions.

Setting Banner Ad Properties

In the SMIL file body, instead of including an <img/> tag link to an ad image file, you use a <RealAdInsert/> tag that triggers automatic ad insertion. In the preceding SMIL example, the ad insertion tag looks like this:

```
<RealAdInsert region="ad_banner" dur="9min" fill="freeze"/>
```

When RealPlayer requests a SMIL file that contains an ad insertion tag, RealServer replaces the <RealAdInsert/> tag with a SMIL <img/> tag that includes a src attribute pointing to a banner ad or, with ad rotation, to a series of banner ads. The <RealAdInsert/> tag can also contain SMIL attributes such as region="ad_banner" and fill="freeze". RealServer simply includes these SMIL attributes in the replacement <img/> tag. You need to specify a duration time only when using rotating banner ads.

Specifying Durations for Rotating Banner Ads

To use rotating banner ads with a live broadcast, do not set a duration time in the <RealAdInsert/> tag. This way RealServer will send new banner ads to RealPlayer as long as it plays the broadcast. For prerecorded content, however, set a duration that is approximately equal to the length of the broadcast. If the requested content is 9 minutes long, for example, you would set the ad duration as dur="9min".

You specify an ad rotation duration when serving prerecorded content so that RealSystem can calculate the presentation length. If you set no duration, RealServer assumes that you want a continuous ad stream for a live broadcast, and it disables RealPlayer's clip position slider. RealServer then sends new ads until the viewer halts the presentation. Although the presentation will play OK, the viewer will not be able to fast-forward or rewind it.

You can set the ad duration shorter than the requested content if you do not want ads to display for the entire length of the presentation. In this case, use a fill="remove" or fill="freeze" attribute in the <RealAdInsert/> tag to specify whether you want the last ad banner to disappear or remain on-screen.

Do not set the ad duration significantly longer than the length of the content, however. RealSystem considers the ad duration when calculating the presentation's total running time. If you set an ad duration of 20 minutes for a 10-minute video, for example, RealPlayer indicates in its status bar that the presentation lasts 20 minutes. If a viewer then moves the RealPlayer clip position slider forward to the 15-minute mark, for instance, the video will stop, possibly leading the viewer to conclude that the presentation is flawed.

Additional Information

See "Specifying Timing" on page 92 for more on SMIL timing values.

Interleaving Media Ads with Clips

Audio, video, and animation are quickly gaining popularity as advertising media, with many ad-serving companies offering media ads in formats such as RealVideo and Flash. With RealSystem's advertising capability, you can play media ads before, between, or within requested clips. To do this, you create a SMIL file that defines when and how ads appear in relation to requested clips. Instead of supplying URLs to ad clips, you write `<RealAdInsert/>` tags that cause RealServer to insert media ads automatically.

Note

The URL for the SMIL file determines the type of media ad inserted in place of `<RealAdInsert/>`. Your RealServer administrator sets up these request URLs and can give you information about media ad sizes, formats, and bandwidths.

Tip

Always account for the higher bandwidth requirements of media ads, and make sure that your target audience has sufficient bandwidth to receive the ad as well as the requested clips. For more on bandwidth, see "Step 3: Develop a Bandwidth Strategy" on page 22.

Creating a SMIL File for a Media Ad

To use media advertising, create a SMIL file that organizes all clips, whether they are ad clips or requested content. For a media ad, use a `<RealAdInsert/>` tag in place of a clip source tag such as `<video/>`. Media ads typically play before or between requested clips. The number, placement, and timing of media ads depend entirely on how you construct the SMIL file.

SMIL File with No Regions

The following SMIL file has no layout. It simply plays two video clips in sequence (the first clip is the media ad):

```
<smil>
  <body>
    <seq>
     <RealAdInsert/>
     <video src="rtsp://realserver.example.com/video/video.rm"/>
    </seq>
  </body>
</smil>
```

SMIL File with Regions

In the preceding example, the lack of SMIL regions causes the RealPlayer display window to resize automatically for each clip. To prevent resizing, or to lay out clips that play together, define SMIL regions for the clips. The following is an example of one video region defined for two clips played in sequence:

```
<smil>
  <head>
    <layout>
      <root-layout width="240" height="180"/>
      <region id="videoregion" width="240" height="180"/>
    </layout>
  </head>
  <body>
    <seq>
      <RealAdInsert region="videoregion"/>
      <video src="rtsp://realserver.example.com/video/video.rm"
         region="videoregion"/>
    </seq>
  </body>
</smil>
```

The `<RealAdInsert/>` tag in the preceding example includes the SMIL `region` attribute to associate the ad with the correct SMIL region. You can include other SMIL attributes, such as the `dur` or `fill` attribute, in a `<RealAdInsert/>` tag as well.

Additional Information

"Laying Out Multiple Clips" on page 99 explains how to set up SMIL regions.

Disabling the RealPlayer Playlist

When a media ad and a requested clip play in sequence, the viewer can skip the ad by choosing the requested clip in the RealPlayer playlist. To disable the playlist through SMIL, enclose the <seq> group within <par> and </par> tags, as in this example:

```
<smil>
  <body>
    <par>
      <seq>
      <RealAdInsert/>
      <video src="rtsp://realserver.example.com/video/video.rm"/>
      </seq>
    </par>
  </body>
</smil>
```

Note

The use of the <par> group also helps reduce the second clip's preroll, as explained in "Smoothing Transitions Between Clips" on page 205.

Additional Information

For more on the RealPlayer playlist, see "Managing the Playlist" on page 119.

Playing Interstitial Ads

By using SMIL's clip-timing commands, you can insert media ads within a requested clip. The following example inserts three media ads into a long video called news.rm. The effect is similar to a television program with commercial breaks:

```
<smil>
  <head>
    <layout>
      <root-layout width="240" height="180"/>
      <region id="videoregion" width="240" height="180"/>
    </layout>
  </head>
  <body>
    <par>
      <seq>
```

```
      <video src="news.rm" region="videoregion" clip-end="8.5min"/>
      <RealAdInsert region="videoregion"/>
      <video src="news.rm" region="videoregion" clip-begin="8.5min"
        clip-end="16.75min"/>
      <RealAdInsert region="videoregion"/>
      <video src="news.rm" region="videoregion" clip-begin="16.75min"
        clip-end="24.5min"/>
      <RealAdInsert region="videoregion"/>
      <video src="news.rm" region="videoregion" clip-begin="24.5min"/>
    </seq>
  </par>
 </body>
</smil>
```

As shown in the preceding example, the three ad clips and the main video clip, `news.rm`, are in a `<seq>` group. The `news.rm` video starts at its actual beginning and plays for 8.5 minutes. It then stops while the first ad plays. The `news.rm` clip then resumes where it left off, stopping again at later intervals for the second and third ads. Because the ad clips include no SMIL timing commands, each one plays from beginning to end without stopping. RealServer delivers a new video ad for each `<RealAdInsert/>` tag.

Additional Information

For more on clip timing, see "Setting Internal Clip Begin and End Times" on page 93.

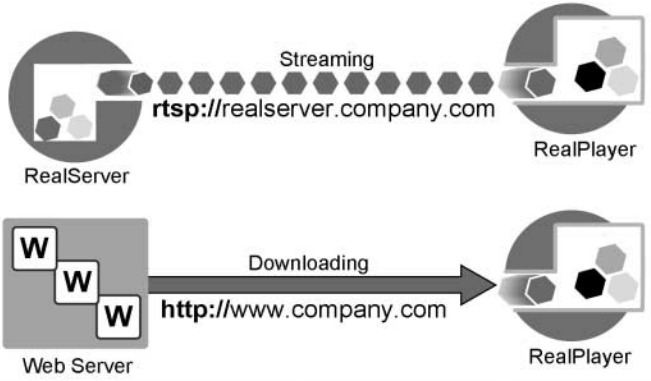
DELIVERING A PRESENTATION

When you finish building your RealSystem presentation, you place the clips on RealServer for streaming. This chapter explains how to link your Web page to your presentation. It also describes how a Web server can play back some RealSystem presentations.

RTSP and HTTP

To deliver HTML pages and graphics, a Web server uses HyperText Transport Protocol (HTTP), as you can see in Web page URLs that begin with `http://`. HTTP downloads files without regard to timelines, making clips with timelines more likely to stall. Although RealServer can also use HTTP, URLs for media clips streamed by RealServer begin with `rtsp://`, which causes it to use RealTime Streaming Protocol (RTSP).

RealServers Stream with RTSP, whereas Web Servers Download with HTTP



Designed specifically for streaming, RTSP enables RealServer to adjust streaming data to keep clips playing smoothly. When two clips play side-by-side, for example, RealPlayer communicates with RealServer about each clip's progress, indicating how much data it needs to keep playback synchronized. RealServer can then adjust the data flow to compensate for changing network

conditions, reducing low priority data if necessary to ensure that crucial data gets through. Communication like this is not possible through HTTP.

Which URLs Use Which Protocol

When you assemble a RealSystem presentation, it's important to understand clearly which URLs should use HTTP and which should use RTSP:

- RTSP in SMIL and Ram files for clips on RealServer

  Use `rtsp://` in URLs in which RealPlayer requests clips from RealServer. These URLs occur in SMIL files (`.smil`) and Ram files (`.ram` or `.rpm`).

- HTTP in SMIL and Ram files for clips on Web servers

  Use `http://` in SMIL and Ram file URLs only if the clips are stored on a Web server instead of on RealServer. Because a Web server does not use RTSP, you cannot use `rtsp://` in a URL to a clip stored on a Web server.

- HTTP in Web pages

  Web page links to a Web server or RealServer always start with `http://`. Web browsers cannot interpret streaming information sent by RealServer through RTSP. The Web browser can connect to RealServer through HTTP, though, because RealServer also uses HTTP.

Additional Information

For more on SMIL file URLs, see "Specifying Clip Locations" on page 85. "URL Reference" on page 202 will help you remember when to use RTSP URLs vs. HTTP URLs.

Streaming Clips from RealServer

RealServer is the preferred host for RealSystem presentations. Designed specifically to stream multimedia over networks, RealServer keeps multiple clips synchronized and uses many advanced features to ensure that clips stream smoothly, even under adverse network conditions.

When you stream clips from RealServer, the RealServer administrator creates content directories and tells you the basic URLs to use. The administrator can also set up features such as password authentication and pay-per-view. When your media clips and SMIL file are ready, transfer them to RealServer and place them in the directories prepared by the administrator. Then link your

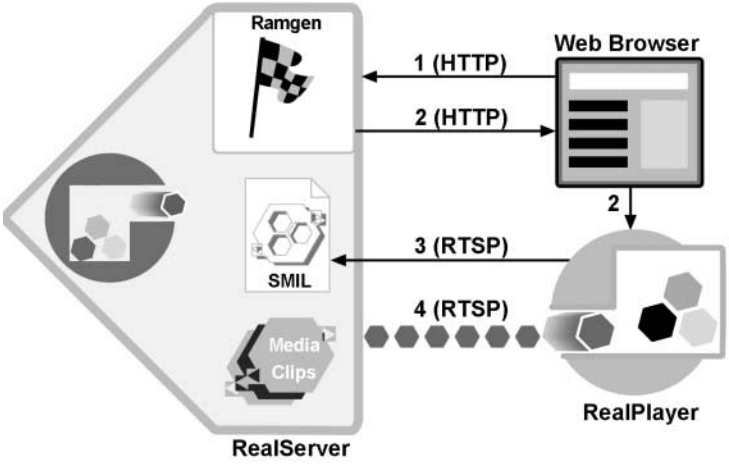
Web page to your presentation as described in "Linking Your Web Page to RealServer" on page 168.

Tip

RealProducer and RealSlideshow can transfer files to RealServer automatically. Refer to their manuals or online help for more information.

With RealServer, you can use Ramgen to launch RealPlayer automatically, eliminating the need to write a separate Ram file. Your Web page URL simply points to your media clip or SMIL file on RealServer and includes a `ramgen` parameter. The following illustration shows the process of requesting a presentation through Ramgen. This example uses a SMIL file that coordinates multiple clips, but you can also link to a single clip directly without using SMIL.

Requesting a Presentation from RealServer Using Ramgen



1. Using HTTP, the Web browser requests the SMIL file from RealServer. The URL includes a parameter that invokes Ramgen.
2. RealServer's response causes the Web browser to launch RealPlayer as a helper application and to give it the URL to the SMIL file.
3. RealPlayer requests the SMIL file from RealServer using RTSP.
4. With the information in the SMIL file, RealPlayer requests and receives the streaming media clips.

Linking Your Web Page to RealServer

With your clips on RealServer, link your Web page to the SMIL file by using an HTML hypertext link that looks like the following:

```
<a href="http://realserver.example.com:8080/ramgen/media/sample.smil">...</a>
```

If the presentation plays back directly in the Web page through RealPlayer's Netscape plug-in, the URL occurs within an `<EMBED>` or `<OBJECT>` tag and looks like this:

```
SRC="http://realserver.example.com:8080/ramgen/media/sample.smil?embed"
```

The following table lists and describes the components of these URLs. Contact your RealServer administrator to get the actual RealServer address, HTTP port, and Ramgen directory structure.

URL Components in a Web Page Link to RealServer

URL Component	Meaning
`http://`	This causes the browser to contact RealServer through HTTP. (Web browsers do not use RTSP.)
`realserver.example.com`	This address varies for each RealServer. It typically uses an identifier such as `realserver` instead of `www`. It may also use a numeric TCP/IP address, such as `204.71.154.5`.
`:8080`	This is the port RealServer uses for HTTP connections. Separate the port and address with a colon. You can leave the port number out if RealServer uses port `80` for HTTP connections. Include the port number if RealServer uses any port other than 80 for HTTP.
`/ramgen/`	As "Using Ramgen" explains, this parameter launches RealPlayer without the use of a separate Ram file.
`/media/`	Following `/ramgen/`, the URL may list other directories, depending on where the clip resides on RealServer.
`sample.smil`	This is the SMIL file for your presentation. If you have only one clip to stream, you can link directly to that clip instead of to a SMIL file.
`?altplay=`*`file.ext`*	This Ramgen option specifies an alternate presentation created for earlier versions of RealPlayer. See "Listing Alternate Presentations with Ramgen" on page 169.
`?embed`	This Ramgen option embeds the presentation in a Web page. See Chapter 8 for complete information on Web page playback.

Using Ramgen

In your Web page hyperlink, the /ramgen/ parameter shown in the preceding examples causes the Web browser to launch RealPlayer without the use of a separate Ram file. This parameter designates a virtual directory in RealServer, and can be followed in the URL by actual directory listings. If your RealServer does not use Ramgen, you can write a Ram file as explained in "Creating a Ram File Manually" on page 173. A Ram file also enables you to use some RealPlayer features, such as playing a clip at double or full-screen size.

Listing Alternate Presentations with Ramgen

With altplay, you can use a single link to stream new clips to newer versions of RealPlayer, and to stream older clips to earlier versions of RealPlayer. Suppose that you have a RealVideo 5 clip and a RealVideo 8 clip laid out using SMIL. You link to the SMIL file using Ramgen as described in the preceding section, and you include altplay to list the older clip:

```
<a href="http://.../ramgen/media/sample.smil?altplay=old_sample.rm">
```

This link instructs RealServer to point RealPlayer G2 or later to sample.smil. Earlier versions of RealPlayer receive the URL to old_sample.rm. RealServer uses the streaming protocol appropriate for each RealPlayer version, whether RTSP or the older PNA. Note that altplay specifies the clip, not a Ram file. Because of this, the older clip must reside in the same directory as the new content.

Combining Ramgen Options

The question mark operator (?) separates Ramgen options from the main URL. To use multiple Ramgen options, you use a question mark before the first option and separate the remaining options with ampersands (&). The order of options does not matter. For example, the following link uses altplay and embed:

```
<a href="http://.../ramgen/media/sample.smil?embed&altplay=old_sample.rm">
```

You can use the ? operator to include earlier Ram file options when using altplay. If your Ram file URL for a RealVideo 5 clip specified an end time, for example, include that option in the Ramgen URL after altplay. The following example shows an end time set for old_sample.rm:

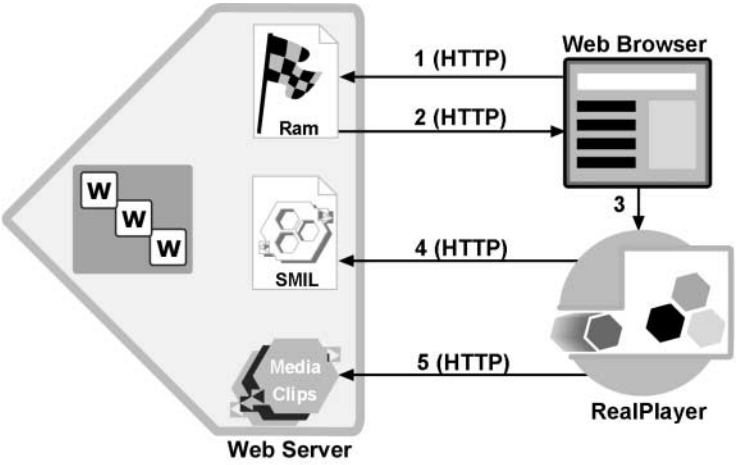
```
<a href="http://.../ramgen/media/sample.smil?altplay=old_sample.rm&end=7:45">
```

Playing Clips from a Web Server

If you do not have access to RealServer, you can host your presentation on a Web server. Although not as robust as RealServer streaming, Web server playback provides a reasonable method for sending simple presentations to a small number of users. This type of playback is not recommended for lengthy or complicated presentations, however, or for clips viewed simultaneously by large groups.

A Ram file launches RealPlayer when the presentation plays back from a Web server. You therefore need to write the Ram file (as described in "Creating a Ram File Manually" on page 173) and place it on the Web server. The following illustration shows the process of requesting a presentation from a Web server. All network activity uses HTTP.

Requesting a Presentation from a Web Server



1. The Web browser requests the Ram file from the Web server.
2. The Web server downloads the Ram file to the browser.
3. The Ram file extension (`.ram` or `.rpm`) causes the Web browser to launch RealPlayer. Note that the `.ram` extension launches RealPlayer as a separate application, whereas the `.rpm` extension plays the presentation within the Web page.

Additional Information

For more information on embedding a presentation in a Web page, see Chapter 8.

4. RealPlayer receives the Ram file and requests the SMIL file from the Web server. Alternately, the Ram file can simply list either a single clip or multiple clips played in sequence.
5. With the information in the SMIL file, RealPlayer requests and receives the clips from the Web server.

Limitations on Web Server Playback

Because Web servers are not designed to manage bandwidth or keep multiple clips synchronized, streaming clips delivered by a Web server are more likely to stall than are clips streamed by RealServer. To ensure that a presentation hosted by a Web server plays as smoothly as possible, observe the following points.

No SureStream Clips Encoded for Multiple Bandwidths

A Web server cannot send just one stream from a SureStream clip encoded for several bandwidths. Instead, it downloads the entire clip, causing a very high preroll. You must therefore encode each RealAudio or RealVideo clip for just one bandwidth. When using RealProducer, select the option for Web server playback and choose your target audience. To support multiple bandwidths, encode separate clips for various bandwidths and use SMIL to let RealPlayer choose which clip to play.

Additional Information

For more on using SMIL to list clip choices, see "Setting Bandwidth Choices" on page 110.

No Secure RealAudio and RealVideo Clips

When you encode RealAudio and RealVideo clips with RealProducer, you have an option to prevent RealPlayer users from recording the streamed clips to their computers. This feature works only when RealServer streams the clips. When a Web server delivers the clips, users still cannot record the clips through RealPlayer, but their Web browsers will cache the clips. Additionally, any user can click on your Web page hypertext links and use **Save as...** commands to download the clips from the Web server.

Limited Ability to Keep Parallel Clips Synchronized

A Web server does not consider clip timelines when downloading data. Nor does it receive feedback from RealPlayer about the presentation's progress.

Web server playback therefore makes it harder for RealPlayer to keep clips synchronized. A presentation that plays large clips in parallel may stall when the RealPlayer connection has little bandwidth to spare.

RealPix Presentations Require Clip Size Information

RealServer determines when to stream each RealPix image based on the image's place in the presentation timeline. Because a Web server cannot do this, you must indicate each image's file size in the RealPix markup. This enables RealPlayer to calculate when to request an image from the Web server so that all image data has arrived by the time the image displays. If the file size information is missing, RealPlayer requests all images when the presentation starts, causing a high preroll.

SMIL File Optional

When delivering a single clip or a few clips played in sequence, you do not need a SMIL file. Instead, you can simply list the clips in order when writing your Ram file, as described in "Creating a Ram File Manually" on page 173. However, you can also have your Ram file specify a SMIL file that lists the clip locations, creates a layout, times the presentation, and so on.

SMIL Internal Timing Commands Do Not Work

Although you can use SMIL to lay out and time your presentation, you should not use the `clip-begin` and `clip-end` attributes. A Web server cannot begin to download a clip at a certain point in its timeline. With `clip-begin="5min"`, for example, RealPlayer must wait until it has received the first 5 minutes of clip data before it can play the clip. This results in an unacceptably long wait.

Additional Information

"Setting Internal Clip Begin and End Times" on page 93 describes these SMIL commands.

No Ad Insertion

The RealSystem ad insertion feature, described in Chapter 9, does not work with Web servers. Only RealServer can replace `<RealAdInsert…/>` tags in SMIL files with URLs to ad clips.

No RealPlayer Seeking

Because a Web server cannot jump to a new position in a clip's timeline, the RealPlayer position slider cannot fast-forward the clip. If the user moves the

slider forward, playback pauses as the clip continues to download at its normal rate. RealPlayer resumes playback once the clip data reaches the specified timeline position.

No RTSP URLs

Because Web servers do not support RTSP, all URLs in presentations hosted by Web servers should begin with `http://`. This includes all URLs in a SMIL file or Ram file.

No Live Broadcast

Live broadcast is not possible because Web servers can download only clips that are stored on disk.

Configuring Web Server MIME Types

To download a RealSystem presentation from a Web server, the server must be configured with the MIME types listed in the following table. If you are using an ISP, ask the Web server administrator to configure the MIME types for you.

Web Server MIME Types for RealSystem Files

File Type	Extension	MIME Type
Ram	.ram	audio/x-pn-realaudio
Embedded Ram	.rpm	audio/x-pn-realaudio-plugin
SMIL	.smil and .smi	application/smil
RealAudio	.ra	audio/x-pn-realaudio
RealVideo	.rm	application/x-pn-realmedia
Flash	.swf	application/x-shockwave-flash
RealPix	.rp	image/vnd.rn-realpix
RealText	.rt	text/vnd.rn-realtext

Creating a Ram File Manually

A Ram file is a text file with the extension .ram (.rpm for playback in a Web page). When a browser receives this file, it launches RealPlayer as a helper application. RealPlayer then requests the clips listed in the Ram file. As described in “Using Ramgen” on page 169. RealServer can launch RealPlayer

without using a Ram file. When doing any of the following, however, you may need to write a Ram file:

- Streaming from a RealServer not set up to use Ramgen.
- Hosting a RealSystem presentation on a Web server.
- Playing back clips that reside on the user's local computer.
- Opening a clip in a specific RealPlayer mode, such as full-screen mode.

➤ To create a Ram file:

1. Open any editor or word processor that can save files as plain text. On the top line, enter the full URL of the SMIL file or the media clip. You may need to verify URLs with your RealServer or Web server administrator. As you can see in the following scenarios, URLs vary with the playback context:

   - RealServer streaming

     The following example links to a SMIL file on a RealServer that does not use Ramgen:

     ```
     rtsp://realserver.example.com/media/sample1.smil
     ```

     To deliver a few clips (but not SMIL files) in sequence, list the URLs in their playback order as shown here:

     ```
     rtsp://realserver.example.com/media/video1.rm
     rtsp://realserver.example.com/media/video2.rm
     rtsp://realserver.example.com/media/video3.rm
     ```

   - Web server playback

     For Web server playback, you specify HTTP and the Web server name, along with the SMIL file or media clip:

     ```
     http://www.example.com/media/video1.rm
     ```

     To deliver a few clips (but not SMIL files) in sequence, list the URLs in their playback order:

     ```
     http://www.example.com/media/video1.rm
     http://www.example.com/media/video2.rm
     http://www.example.com/media/video3.rm
     ```

   - Local playback

     For local playback of clips residing on the user's computer, start the URL with `file://` and list clips in their locations relative to the Ram

file. For instance, the following example specifies a clip that resides one level below the Ram file in the `media` directory:

```
file://media/video1.rm
```

To deliver a few clips (but not SMIL files) in sequence, list the URLs in their playback order:

```
file://media/video1.rm
file://media/video2.rm
file://media/video3.rm
```

Additional Information

For more information on general URL syntax, see "Specifying Clip Locations" on page 85.

2. For a presentation played back from RealServer, you can support earlier versions of RealPlayer (such as RealPlayer 4 or 5) just as Ramgen does with the `altplay` option. To do this, add the marker `--stop--` after the RTSP URL and then specify the URL for the older clip just as it appeared in your previous Ram file. Here's an example:

   ```
   rtsp://realserver.example.com:554/media/sample.smil
   --stop--
   pnm://realserver.example.com:7070/media/old_sample.rm
   ```

   The second URL specifies the older RealSystem protocol with `pnm://` and designates RealServer's PNA port. When RealPlayer connects, it chooses the URL based on its favored protocol. For this reason, you cannot list two URLS that both use the same protocol, whether `rtsp://`, `pnm://`, or `http://`.

3. Save the Ram file as plain text with a `.ram` extension (played in RealPlayer) or a `.rpm` extension (played in a Web browser).

4. Move your Ram file to RealServer or your Web server. Even if all your media clips are on RealServer, you can place the Ram file on your Web server. When the browser receives a Ram file, it turns it over to RealPlayer, which uses the URLs in the file to request clips. Hence the Ram file and the media clips do not need to reside on the same computer.

5. For `.ram` files, link your Web page to the Ram file by using an HTML hyperlink such as this:

   ```
   <a href="http://www.example.com/media/sample.ram">click for video</a>
   ```

For .rpm files, incorporate the link URL in the <EMBED> or <OBJECT> tag as described in "Setting Basic Parameters" on page 139. If the Ram file is on RealServer, the URL must not use the ramgen parameter.

Setting a Presentation's Starting Mode

In the Ram file, you can control how RealPlayer initially displays a clip or SMIL presentation. You can play a clip back at double its normal size, for example, or open RealPlayer in its compact mode. To set the starting mode, add one of the following options to the end of the Ram file URL:

?screensize="double"	Opens the clip or presentation at double its normal size.
?screensize="full"	Starts the clip or presentation in full-screen mode. This hides all windows to make the user's monitor look like a television screen.
?screensize="original"	Opens the clip or presentation at its normal size. This is the default behavior if you omit screensize.
?mode="compact"	Opens RealPlayer 8 or later in compact mode so that only the basic controls appear. It does not affect earlier versions of RealPlayer.

This example opens a SMIL presentation in full-screen mode:

```
rtsp://realserver.example.com/media/sample1.smil?screensize="full"
```

The next example opens a RealVideo clip at double its normal size:

```
rtsp://realserver.example.com/media/video1.rm?screensize="double"
```

To include two options in the URL, use a question mark (?) before the first option, and then separate the second option with an ampersand (&). The following example opens a RealVideo clip at double its normal size and sets RealPlayer to its compact mode:

```
rtsp://realserver.example.com/media/video1.rm?screensize="double"&mode="compact"
```

Notes on Setting a Starting Clip Size

If you plan to use double or full-screen mode when playing back a clip or presentation, keep the following points in mind:

- Do not use screensize="full" with mode="compact".
- RealPlayer may not offer full-screen mode on all operating systems. This mode is generally available on Windows computers with DirectX

technology. If RealPlayer for a given operating system does not offer full-screen mode, it plays the presentation at its normal size.

- If RealPlayer offers full-screen mode but has not yet played a clip full-screen, it may first perform a test of this playback mode.
- The double and full-screen modes work best for high-speed clips. They are not recommended for presentations delivered through modems.
- Always test playback when using double and full-screen modes to ensure that the visual quality is acceptable. Some types of clips may not scale well.
- In full-screen mode, the user can control RealPlayer through a context menu accessed by right-clicking (on Windows) or holding down the mouse button (on Macintosh).
- With RealPlayer 7 and later, you can also launch new windows in double and full-screen modes. For more information, see "Popping Up New RealPlayer Windows" on page 127.

Adding Comments to a Ram File

You can add a comment to a Ram file by using a pound sign (#) as the first character on a line. The following example shows two lines commented out of a Ram file:

```
# Two videos and a SMIL presentation
# streamed from RealServer.
rtsp://realserver.example.com/media/video1.rm
rtsp://realserver.example.com/media/video2.rm
rtsp://realserver.example.com/media/sample2.smil
```

Testing Your Presentation

Use the following guidelines to make sure your presentation works well and reaches its target audience:

- Test your presentation in "real world" conditions. If you target 28.8 Kbps connections, for example, request the presentation over a 28.8 Kbps modem.
- Check that the presentation has a preroll (initial buffering) under 15 seconds. After preroll, the presentation should not rebuffer under normal network conditions.

Additional Information
"Buffering" on page 23.

- Verify that video and audio quality is acceptable.
- For a multiclip presentation, verify that clips stay synchronized. Ensure that no stalling occurs because of too many clips playing at the same time or a single clip requiring too much bandwidth. Make sure that clips introduced during a presentation in progress do not stall playback by requiring too much buffering when they start.

  Tip
  If clips introduced during a presentation in progress require too much buffering, see "Smoothing Transitions Between Clips" on page 205.

- Make sure that your presentation works well for an "average" CPU for your audience. For general Web delivery, test playback on both Pentium and Power Macintosh computers with clock speeds around 300 MHz. Do not rely on MMX technology to enhance playback. Not all Web users have MMX computers.

  Tip
  If your presentation is CPU-intensive because it uses complex Flash animation or high-bandwidth video, for example, note this in your Web page.

- When streaming RealAudio clips, ensure that sound quality is acceptable. You may need to experiment with RealAudio codecs to find the best balance between clip bandwidth use and sound quality.
- Test all hypertext links.
- When embedding a presentation in a Web page, verify that the playback window has the correct location and controls.

Advertising on RealGuide

Every day, thousands of people visit RealGuide, RealNetworks' online guide for streaming media sites and live events (**http://realguide.real.com**). If you regularly host streaming media presentations of interest to the public, or if you have a live event you want to advertise, you can submit your listing to

RealNetworks. Simply complete the following online form to list your site or live event:

- **http://realguide.real.com/info/?page=submit**

In the form, you provide the site or event name, the URL, a short description, and the name of a contact person. A RealGuide staff member then verifies your site or event before including it on RealGuide. (RealNetworks reserves the right to edit or refuse submissions.) For live events, please submit your request at least one business day in advance. If you have questions or need to change a listing, please contact RealGuide from this Web page:

- **http://realguide.real.com/info/?page=fbform**

Using RealNetworks Logos

When you create RealSystem content, RealNetworks encourages you to add RealSystem logos to your Web page. You can provide a RealPlayer download link button, for example, so that users can get RealPlayer from RealNetworks' Web site and view your content. You can read RealNetworks' trademark policies and get RealSystem and RealPlayer logos at the following address:

http://www.realnetworks.com/company/guide/index.html

BROADCASTING A PRESENTATION

The Internet is swiftly becoming the next great broadcast medium. RealSystem lets you broadcast live or prerecorded presentations over the Internet or an intranet. This chapter provides background on using broadcast applications and RealServer to broadcast media. Refer to the documentation for RealServer and your tools for instructions on configuring a broadcast.

Tip

Real Broadcast Network™ provides full services for broadcasting small and large events. For details, see **http://www.realnetworks.com/rbn/index.html**.

Broadcasting vs. On-Demand Delivery

When a streaming presentation is delivered on demand, it starts from its beginning when the viewer clicks the presentation link in a Web page. Each viewer can receive the presentation at any time and use RealPlayer's controls to fast-forward or rewind through the presentation.

In a streaming broadcast, however, the user hosting the broadcast starts the presentation at a certain time. Viewers who click the presentation link join the broadcast in progress. Before the broadcast begins and after it ends, the presentation URL is not valid. During the broadcast, the RealPlayer fast-forward and rewind controls do not function.

To make an analogy, on-demand content is like a movie on videotape. The viewer can see it at any time, skip forward, rewind, and pause. A streaming broadcast, on the other hand, is like a movie shown on a television channel. As with a TV broadcast, there are two types of streaming media broadcasts:

- Live content

  Live content is broadcast as it occurs. For example, you can broadcast the output of a video camera across the Internet or an intranet. RealSystem

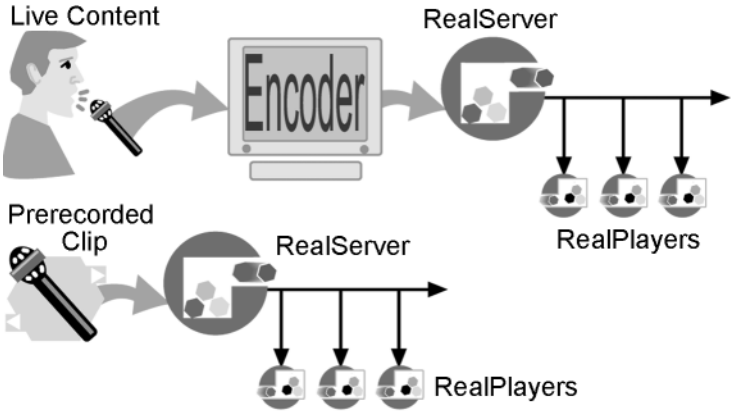
encodes the content as RealVideo in real time without first saving the content as a RealVideo clip.

- Prerecorded content

  Prerecorded content consists of video or audio content you record and save as a digitized file. You can then edit the content before encoding it as a streaming clip and broadcasting it across a network. To the viewer, the presentation looks just like a live broadcast.

The following illustration shows how RealServer delivers both live and prerecorded broadcasts.

Broadcasting Live or Prerecorded Content



Broadcasting Tools

To broadcast a presentation, you need the tools described in the following sections.

Source Capture Equipment

This equipment captures the broadcast content. It is typically a microphone or video camera connected to an audio or video capture card. For text, it could be a live text feed coming in over a network.

Editing Equipment

When you broadcast prerecorded content, you first write the source to a digitized file. You can then use editing software to optimize the file for broadcast. Live content, however, is sent directly from the capture equipment to the broadcast application.

Broadcast Application

A broadcast application takes the live source and encodes it in the appropriate streaming format, sending the output to RealServer. RealProducer, for example, can encode a video camera's RGB or YUV output as RealVideo in real time. A broadcast application typically runs on a separate computer that has a network connection to the RealServer computer. To broadcast prerecorded content, you typically do not need a broadcast application because RealServer can broadcast the clips itself.

RealSystem's open architecture also lets you build a broadcast application to send RealServer any type of data for broadcast. To build such an application, you need the RealSystem Software Development Kit (SDK), available at this Web address:

http://www.realnetworks.com/devzone/downlds/index.html

Sample RealText and RealPix broadcast applications are available in the RealSystem Authoring Kit, available at this Web page:

http://www.realnetworks.com/products/authkit/index.html

RealServer

RealServer streams the broadcast to RealPlayer. The RealServer administrator can give you the broadcast URL and parameters for connecting a broadcast application to RealServer. Because each RealServer has limits on its outgoing bandwidth and the number of streams it can produce, verify that your RealServer has capabilities appropriate for your anticipated audience size.

Additional Information

For information on configuring a broadcast, refer to the *RealServer Administration Guide*, available at **http://service.real.com/help/library/servers.html**.

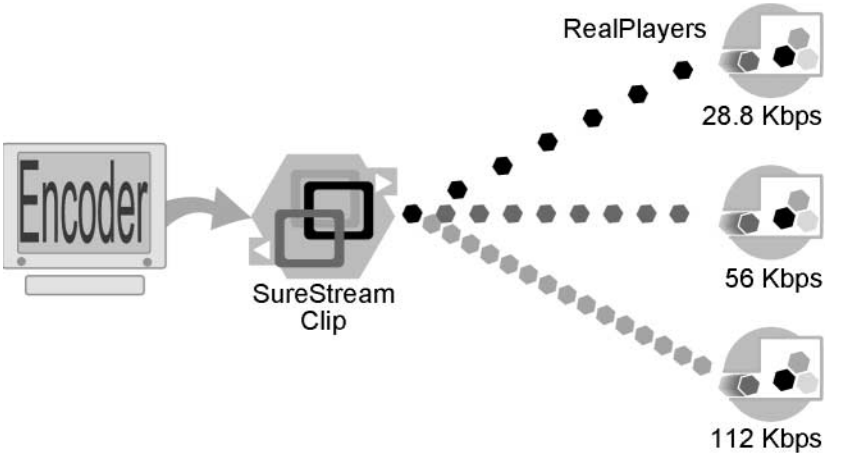
Preparing a Broadcast

This section provides tips for preparing either a live or a prerecorded broadcast. See the manual or online help for your broadcast application for explicit instructions on how to encode and broadcast content. Note that the RealServer administrator sets up RealServer for the broadcast.

SureStream Broadcasts

Using SureStream technology, you can broadcast RealAudio or RealVideo at multiple bandwidths. Each viewer's RealPlayer selects an encoding appropriate for its connection speed. When you begin the broadcast, you specify in RealProducer that you want to use SureStream.

Multiple-Bandwidth Broadcast Through SureStream



To broadcast without using SureStream, you need to have a separate broadcast application running on a separate computer for each bandwidth connection. This is required because older codecs require more CPU power than do SureStream codecs. You then connect each broadcast application to RealServer and broadcast the different streams using different URLs.

Additional Information

For more on codecs, see "Understanding RealAudio" on page 33 and "Understanding RealVideo" on page 47.

Broadcast Computer CPU Power

RealProducer can broadcast a SureStream clip for several different bandwidths in real time using just a moderately powerful PC. Refer to the RealProducer manual or online help for guidelines on computer requirements for broadcasting.

Archiving Broadcasts

Using RealProducer, you can write a live broadcast to a file. RealServer can also archive the broadcast to a file. The latter option may be the better solution if your broadcast is long and your RealProducer computer has limited disk space. The RealServer administrator can set up RealServer to archive the live broadcast.

Broadcast URL and RealServer Parameters

The RealServer administrator can give you the parameters you need to connect a broadcast application to RealServer. The administrator will also provide the broadcast URL or URLs.

Trial Runs

When you broadcast live content, you don't get a second chance. It's good practice to perform a trial run to ensure that the equipment works properly and that the broadcast results are what you expect. Because you can't edit a live broadcast the way you can a prerecorded file, it's important to set your audio levels and plan your video shots carefully in advance.

Additional Information

For pointers on recording audio, see "Capturing Audio" on page 41. For tips on capturing video, see "Recording Video" on page 56.

During both the trial run and the live broadcast, view the broadcast output with RealPlayer. When RealPlayer connects, check that the buffering time for receiving a live stream does not exceed 10 seconds. Throughout the presentation, keep an eye on the broadcast quality. If you experience problems during your trial run, you may need to either reduce the number of SureStream streams or run RealProducer on a more powerful computer.

Advertising an Event

If you are broadcasting content that has wide public appeal, advertise your event on RealGuide (**http://realguide.real.com/**), RealNetworks' online guide to streaming media.

Additional Information

See "Advertising on RealGuide" on page 178.

Using SMIL with a Broadcast

You can use SMIL to coordinate live streams or create a presentation that adds prerecorded content to a live broadcast. This section provides tips on using SMIL with broadcasts.

Additional Information

See Chapter 6 for more on SMIL. Chapter 9 explains how to include ads in broadcasts.

SMIL Limitations for Broadcasts

Within a SMIL file, you treat a broadcast like any other clip. For example, you can assign broadcast streams to SMIL regions and group a broadcast with on-demand clips in a `<seq>` or `<par>` group. However, you cannot use some SMIL features with broadcast streams:

- You cannot replay a broadcast with the SMIL `repeat` attribute.
- The `clip-begin` and `clip-end` attributes do not work with broadcasts. However, you can terminate a broadcast before the stream stops by using the SMIL `end` attribute.

Additional Information

See "Setting Begin and End Times" on page 92.

Including On-Demand Clips with Broadcasts

Using SMIL, you can easily embed a broadcast in a multiclip presentation. SMIL can deliver an on-demand RealPix slideshow along with live RealAudio, for example, when both are in a `<par>` group. It cannot synchronize the on-demand clip with the live stream, however. This is because the on-demand clip's timeline starts when the viewer requests the presentation, whereas the broadcast stream's timeline starts when the broadcast begins.

To illustrate this, suppose that viewer A requests the presentation 2 minutes after the broadcast begins, and viewer B requests it 4 minutes after the broadcast begins. At 10 minutes into the broadcast, both viewers hear the same audio, but viewer A's RealPix clip is at its 8-minute mark, whereas viewer B's clip is at its 6-minute mark. Hence the relationship between the two timelines varies for each viewer.

Synchronizing Multiple Broadcast Streams

If you have multiple broadcast streams for a single presentation, you can synchronize the streams by using SMIL. This compensates for the fact that you typically cannot start two broadcast applications at exactly the same time. For example, you might start a live RealText encoding application a few seconds before a live RealVideo stream. When it receives the streams, RealPlayer delays playing the RealText stream until it can synchronize it with the RealVideo stream.

Tip

RealPlayer synchronizes broadcast streams to audio. For this reason, start your audio/video broadcast application last. This way, RealPlayer will have received and buffered the other streams by the time it receives the audio stream.

The broadcast streams are timestamped according to the broadcast computers' internal clocks. You should therefore run broadcast applications either on the same computer or on different computers that have synchronized clocks. To make RealPlayer synchronize streams, add a `?wallclock=`*`name`* option to the broadcast source tag URLs in the SMIL file, as shown in the following example:

```
<par>
  <video src="rtsp://realserver.example.com/encoder/video.rm?wallclock=sync"/>
  <textstream src="rtsp://realserver.example.com/live/text.rt?wallclock=sync"/>
</par>
```

All `wallclock` attributes should use the same name value (`sync` in the example above), which can be any name you choose. You do not need the `wallclock` attribute for an on-demand clip or a single broadcast stream.

Warning

The `wallclock` attribute works with all RealNetworks clip types, including RealAudio, RealVideo, RealPix, and RealText. It may not function with other clip types broadcast by RealServer, however.

QUICK ANSWERS TO COMMON QUESTIONS

This appendix answers often-asked questions about producing streaming media clips for RealSystem. It also provides URLs for Web sites where you can find tools and helpful information about developing streaming media presentations.

Creating Streaming Clips

RealProducer is the basic tool you use to create clips. Both the *RealProducer User's Guide* and the product's online help guide you through the encoding process. This production guide provides background information and tips on creating high-quality streaming media.

How do I make streaming audio and video clips?

You start with an audio or video source file in a digitized format on your computer. You then use RealProducer's encoding wizard to select the file and set encoding options. The encoding process creates a new streaming clip, leaving the source file unchanged.

Can I encode RealVideo directly from a video camera?

Yes. RealProducer accepts live video input from a camera and live audio input from a microphone. The camera and microphone connect to an audio/video capture card on your computer. RealProducer's encoding wizard then lets you select the live input as the source. In this case, you go directly from live input to encoded clip without creating a digitized source file.

How do I ensure the best quality for streaming clips?

Quality starts at the source. You need high-quality video and audio input for RealProducer to create high-quality streaming clips. Chapter 3 and Chapter 4 include tips on producing good audio and video, respectively. If you are new to media production, learn your editing hardware and software thoroughly,

paying close attention to the manufacturers' recommendations for producing high-quality media files.

What other clips can I stream?

In addition to audio and video, RealSystem can stream the following types of clips:

- Macromedia Flash animation
- GIF, JPEG, and PNG images
- RealPix clips for streaming slideshows
- RealText clips for streaming text

Getting Production Tools

To produce streaming media clips, you need audio and video production tools as well as RealProducer to handle the encoding.

What audio and video editing tools can I use?

You can use any hardware or software designed for capturing and editing audio or video. The digitized output must be in a format that RealProducer accepts, however. Some video editing programs save digitized video in a proprietary format that RealProducer cannot read. However, these programs typically let you export the video to a common format that RealProducer accepts, such as AVI, QuickTime, or MPEG.

> **Tip**
> You can purchase hardware and software for capturing and editing audio or video from RealStore at **http://www.realstore.com**.

What digitized audio and video formats does RealProducer accept as input?

RealProducer accepts many common audio and video formats. These may vary by operating system, though. RealProducer on Macintosh accepts the formats widely used on the Macintosh, such as QuickTime, whereas RealProducer on Windows or UNIX supports the formats widely used on those operating systems. Check the RealProducer manual for your operating system for a list of accepted formats. Information is also available at the following Web page:

http://www.realnetworks.com/products/producer/freevsplus.html

Where can I get RealProducer?

RealNetworks makes versions of RealProducer for Windows 95/98/NT/2000, Macintosh, and Linux. You can download the free version or purchase RealProducer Plus at RealNetworks' Web site:

http://www.realnetworks.com/products/index.html

How do I create a streaming slideshow from still images?

Using RealSlideshow's graphical interface, you can create streaming RealPix presentations from still images. You can even add a soundtrack, or record a narration for each image. You can download RealSlideshow from this Web address:

http://www.realnetworks.com/products/index.html

You can also create RealPix presentations by hand with the RealPix markup language, which is described in *RealPix Authoring Guide,* available at this Web page:

http://service.real.com/help/library/encoders.html

How do I create streaming Flash animation?

You create animation with Macromedia Flash. You can develop animations with Flash 2, 3, or 4. Chapter 5 provides tips for making Flash animation stream well with RealSystem. It doesn't explain how to create Flash animations, however. You can learn more about Flash at from Macromedia's Web site:

http://www.macromedia.com/software/flash/

What's the RealSystem Authoring Kit?

The RealSystem Authoring Kit bundles several RealNetworks tools and manuals into one archive that you can download. This gives you one source for the basic tools and information you need to create streaming media clips. The Authoring Kit is available free at:

http://www.realnetworks.com/products/authkit/

Using SureStream

SureStream provides advanced streaming technology for RealSystem. For more information about SureStream, read "SureStream RealAudio and RealVideo" on page 27.

What is SureStream?

SureStream is a technology that lets a single RealAudio or RealVideo clip stream at different bit rates. It does this by bundling into a single clip multiple streams, each of which runs at a different bit rate. You can make a SureStream clip that streams at either 28.8 Kbps or 56 Kbps, for example. When users request the clip, they automatically receive the stream that best matches their RealPlayer connection speed.

How do I make a SureStream clip?

Using RealProducer, you can choose to use SureStream when you encode audio or video input. The number of SureStream streams you can encode in the clip depends on the type of RealProducer you use. RealProducer Basic encodes two speeds per clip, whereas RealProducer Plus encodes up to eight speeds per clip.

Can I use SureStream with a Web server?

No. A SureStream clip has several streams encoded in a single clip. Unlike RealServer, a Web server cannot extract a specific stream to send to RealPlayer. If you plan to deliver clips from a Web server, you need to set RealProducer to use single-rate encoding.

Writing SMIL Files

Chapter 6 explains the basics of SMIL. Chapter 7 covers the RealNetworks extensions to SMIL. Appendix D is a quick reference you can use once you are comfortable with SMIL.

What is SMIL?

Pronounced "smile," SMIL stands for "Synchronized Multimedia Integration Language." It is an industry-standard markup language used to lay out and time streaming media presentations. SMIL works for RealPlayer the way HTML works for a Web browser.

Is it necessary to use SMIL?

Not always. When you want to stream just one clip, such as a single RealVideo clip, you don't need to use SMIL. You just link your Web page to the clip through a Ram file. For more information, see "What is a Ram file?" on page 194.

When should I use SMIL?

When you stream multiple clips, SMIL gives you the means to lay out the presentation and time its clips. It also provides other features, such as letting you create hyperlinks that start a new presentation. For a rundown of basic SMIL features, see "Understanding SMIL" on page 81.

How do I write SMIL?

SMIL is a simple markup language that you can write with a word processor or text editor. Some software tools (RealSlideshow, for example) create SMIL files automatically.

Streaming Clips

You can stream clips yourself with RealServer, through a service provider that has RealServer available, or, in some cases, from a Web server.

Do I need to have RealServer in addition to RealProducer?

Not necessarily. RealServer streams the clips created by RealProducer. To run RealServer, you need a computer connected to an intranet or one that has a direct presence on the Internet. You cannot run RealServer if you use an Internet service provider (ISP) to connect to the Internet. If you use an ISP, check whether they have RealServer and whether they can host your streaming presentations for you.

What operating systems does RealServer run on?

RealServer runs on Windows NT/2000 and many UNIX platforms, including Linux. For a list of available platforms, visit RealNetworks' technical support Web site at **http://service.real.com**.

Where do I get RealServer?

RealServer is available on the RealNetworks Web site at **http://www.realnetworks.com/products/index.html**. RealServer Basic is free.

Can I stream clips from a Web server instead of RealServer?

Sometimes. A Web server can deliver many types of clips, including RealAudio and RealVideo. There are limits to Web server delivery, however. If you plan to use a Web server for clip delivery, read "Limitations on Web Server Playback" on page 171 first.

What is a Ram file?

A Ram file, also called a "metafile," is a simple text file with the extension `.ram` that typically consists of just one line—the URL to a streaming presentation. Your Web page does not link directly to your presentation. Instead, it links to the Ram file, which ensures that RealPlayer launches. RealPlayer then uses the URL in the Ram file to request the presentation. "Creating a Ram File Manually" on page 173 explains how to write a Ram file.

> **Tip**
> When you stream clips with RealServer, you can eliminate the Ram file by using the Ramgen utility. For more information, see "Using Ramgen" on page 169.

If I use SMIL, do I need a Ram file?

Yes. The SMIL file lists the URLs for clips. The Ram file supplies RealPlayer with the URL to the SMIL file (or to your streaming clip, if you're not using SMIL). The Ram file is always necessary because its `.ram` extension launches RealPlayer.

Can I place clips in a Web page?

Yes. Chapter 8 explains how to embed clips and RealPlayer controls in a Web page. RealPlayer still plays the clips, but it does so "behind the scenes" as a browser plug-in rather than by launching separately.

Why does RealServer use RTSP rather than HTTP?

Web servers use HTTP to deliver Web pages and graphics. HTTP is designed to download small files quickly and efficiently. It is not suited for streaming large

media clips, though. RTSP, which stands for "RealTime Streaming Protocol," is an industry-standard protocol that overcomes the deficiencies of HTTP for streaming media. RTSP enables RealServer and RealPlayer to stream long clips and compensate for changing network conditions.

How do I stream clips with RTSP?

When a clip resides on RealServer, make sure that the URL used to request it starts with `rtsp://` rather than `http://`. An RTSP URL must be in a file read by RealPlayer, such as a Ram file or a SMIL file. It cannot be in an HTML hyperlink, because a Web browser does not know how to make an RTSP request. For more on this, see "RTSP and HTTP" on page 165.

Advertising

Chapter 9 explains how to stream ads using SMIL. The RealServer administrator performs most of the work for setting up ad streaming, as described in *RealServer Administration Guide*.

How does advertising with RealSystem work?

Ad streaming uses `<RealAdInsert/>` tags in SMIL files to designate when and where ads appear in presentations. When RealServer serves a SMIL file, it replaces the tags with URLs to ad files. These URLs come from a separate ad server, and the RealServer administrator can configure RealServer to work with virtually any ad-serving system.

What kinds of ads can I stream?

RealServer can stream banner ads in image formats such as GIF, JPEG, and PNG. It can display one ad per presentation, or it can make new ads appear at specific intervals during the presentation. RealServer can also stream ads in RealAudio, RealVideo, and Flash formats. You can have ads either precede the requested clips or appear as commercial breaks during the clips.

How do I set up advertising?

You need RealServer with the optional advertising extension installed. You also need either to have an ad server, or to sign up with an online ad provider such as DoubleClick. The RealServer administrator configures RealServer to work with your ad-serving system, determining what types of ads are to be

streamed. You then write SMIL files with `<RealAdInsert/>` tags that specify where ads are to be placed.

Do I have to write SMIL files to stream ads?

No. Writing your own SMIL files gives you a more flexible way to stream ads, but RealServer can automatically include ads with requested clips or SMIL presentations. If your RealServer hosts a large number of RealAudio or RealVideo clips, for example, you can simply let RealServer lay out an ad for each clip.

Broadcasting

For complete information about broadcasting streaming media, see *RealProducer User's Guide*, *RealServer Administration Guide*, and Chapter 11 of this production guide.

What do I need for broadcasting over a network?

You need the following:

- An audio or video capture card on your computer, to digitize the input from a microphone or camera.
- RealProducer on the same computer as the capture card, to encode the output in a streaming format and send the stream to RealServer.
- RealServer, to broadcast the stream to one or more RealPlayers. RealServer typically does not run on the same computer as RealProducer.

Can I broadcast through my ISP?

Possibly. If you connect to the Internet through an ISP, you may be able to broadcast streaming media, provided that your ISP has RealServer available and offers broadcasting services. To do this, you will need a fast Internet connection to your ISP. You cannot broadcast through an ISP by running RealServer on your home computer.

Can I use SureStream in a broadcast?

Yes. Using SureStream is recommended because it ensures that users connecting at different speeds will each receive the best possible stream. You need to make sure, however, that the computer running RealProducer has enough power to encode all the SureStream streams at the same time. Check

RealProducer's manual or online help for system requirements, and perform a trial run before streaming the actual broadcast.

Can I broadcast with a Web server instead of RealServer?

No. You need RealServer to broadcast streaming presentations. Web servers are designed to serve HTML pages and graphics to different users at different times. They are not designed to broadcast the same presentation to multiple users simultaneously.

Does a broadcast have to be live?

No. "Broadcasting" means to send out a stream that more than one RealPlayer user can view at the same time. The broadcast can be live, meaning that the input originates from a microphone or video camera. Or it can be prerecorded, meaning that it originates from a digitized clip prepared in advance. If it's prerecorded, you don't need to use RealProducer during the broadcast. You just put the clip on RealServer and then set up RealServer to broadcast the clip as a simulated live event.

Can I use SMIL with a broadcast?

Yes. You can use SMIL to include ads with the broadcast, or deliver static clips alongside the broadcast. In the SMIL file, you simply treat the broadcast as a static clip. The only difference is that you use a special URL created by the RealServer administrator that identifies the resource as a broadcast rather than a clip.

How many people can I reach with a broadcast?

That depends entirely on your RealServer and the network bandwidth it has available. Each RealServer can broadcast to a specific number of RealPlayers, as defined in its license agreement. For large broadcasts, you can use a network of RealServers to reach thousands of RealPlayers.

Can RealNetworks broadcast clips for me?

Yes. Real Broadcast Network (RBN) offers a wide range of services for hosting broadcasts. Learn more about RBN at:

http://www.realnetworks.com/rbn/index.html

Getting Technical Support

RealNetworks offers a range of technical support features and documentation.

How do I get technical support from RealNetworks?

RealNetworks Technical Support operates an extensive Web site at **http://service.real.com**. The site includes answers to frequently asked questions, a documentation library, and a searchable knowledge base. To place a service call with Technical Support, fill out the e-mail form at the following Web page:

http://customerrelations.real.com/scripts/rnforms/contact_tech_service.asp

Where can I find additional documentation?

RealNetworks Technical Support maintains a documentation library at **http://service.real.com/help/library/index.html**. Most documents are available as bundled HTML archives that you can download, uncompress, and read with a Web browser. Many documents are also available in PDF format, which is suitable for printing.To read PDF files, you need Adobe's Acrobat Reader, which is available from Adobe's Web site:

http://www.adobe.com/products/acrobat/readstep.html

Where should I go for the latest RealSystem information?

The RealNetworks DevZone is the main information site for content authors and software developers working with RealSystem products. You can find it at the following Web address:

http://www.realnetworks.com/devzone

QUICK STEPS FOR STREAMING CLIPS

This appendix provides examples that show how to set up simple streaming presentations quickly. Once you are familiar with the basic steps for hosting RealSystem presentations, you can easily create more complex presentations.

Note

You will need to get the correct URLs for media clips from the RealServer or Web server administrator. Substitute those values for the URLs given in the following examples.

Streaming a RealAudio or RealVideo Clip

It's simple to add a RealAudio or RealVideo clip to your Web page. Following the instructions below, you can either stream the clip from RealServer or download it from a Web server. Either way, the clip plays back in RealPlayer.

Note

Before using a Web server, read "Limitations on Web Server Playback" on page 171.

➤ To create the clip:

1. Prepare your audio or video source file for encoding. This can include normalizing the audio source file, for example, or setting the video's window size.

   Additional Information

   See "Optimizing Audio" on page 43 and "Digitizing Video" on page 58.

2. Use RealProducer to encode the RealAudio or RealVideo clip from your audio or video source file. Both RealVideo and RealAudio clips use the file extension `.rm`.

Note

RealProducer Basic and RealProducer Plus are available at **http://www.realnetworks.com/products/index.html**.

➤ To stream the clip from RealServer using Ramgen:

1. Transfer the clip to the RealServer directory prepared by the RealServer administrator.
2. Link your Web page to the clip with an HTML hyperlink that begins with `http://`, and that specifies the RealServer address along with the `ramgen` parameter. You can get this information from the RealServer administrator. In your HTML source file, the link will look like the following example:

```
<a href="http://realserver.example.com:8080/ramgen/videos/myclip.rm">
Click here</a> to see my RealVideo presentation.
```

Additional Information

See "Streaming Clips from RealServer" on page 166.

3. In your Web browser, click the link to verify that it works. RealPlayer will launch as a helper application and, after a few seconds of buffering, will play the streaming clip.

➤ To play the clip back from a Web server:

1. With any text editor, open a new file and enter the URL your clip will have on the Web server, as in the following example:

```
http://www.example.com/media/myclip.rm
```

2. Save this file as plain text with the file extension `.ram`. This is your Ram file.

Additional Information

See "Creating a Ram File Manually" on page 173.

3. Transfer the media clip and the Ram file to the appropriate directory on the Web server.
4. Link your Web page to the Ram file (not the media clip) with a standard HTML hyperlink such as the following:

```
<a href="http://www.example.com/media/myclip.ram">
Click here</a> to see my RealVideo presentation.
```

5. In your Web browser, click the link to verify that it works. RealPlayer will launch as a helper application and, after a few seconds of buffering, will play the clip.

Additional Information

See "Playing Clips from a Web Server" on page 170.

Embedding a RealVideo Clip in a Web Page

Using RealPlayer's Netscape plug-in, you can embed a RealVideo clip directly in your Web page. The following procedure assumes that the video is 176 pixels wide by 132 pixels high. It places the video window and the full RealPlayer control panel in your Web page.

Additional Information

See "Chapter 8: Playing Clips in a Web Page", beginning on page 137.

➤ To embed the clip in your Web page:

1. In your Web page, add the `<EMBED>` tag with the RealVideo URL, window size, and `ImageWindow` control. The following example assumes that RealServer will stream the presentation:

```
<EMBED WIDTH=176 HEIGHT=132
SRC="http://realserver.example.com:8080/ramgen/videos/myclip.rm?embed"
CONTROLS=ImageWindow CONSOLE=one NOJAVA=true>
```

   If you intend to play the clip back from a Web server, use an HTTP URL and link to a Ram file with the extension `.rpm`, as in the following example:

```
<EMBED WIDTH=176 HEIGHT=132
SRC="http://www.example.com/media/myclip.rpm"
CONTROLS=ImageWindow CONSOLE=one NOJAVA=true>
```

2. You can then add RealPlayer controls by inserting additional `<EMBED>` tags that all use the same URL for the `SRC` parameter. The following example, which assumes that RealServer will stream the presentation, embeds the full RealPlayer control panel in the Web page, linking it to the image window through the same console:

```
<EMBED WIDTH=375 HEIGHT=100
SRC="http://realserver.example.com:8080/ramgen/videos/myclip.rm?embed"
CONTROLS=All CONSOLE=one NOJAVA=true>
```

As in the following example, use an <EMBED> tag when playing the clip back from a Web server:

```
<EMBED WIDTH=375 HEIGHT=100
SRC="http://www.example.com/media/myclip.rpm"
CONTROLS=All CONSOLE=one NOJAVA=true>
```

Tip

Because you can place each <EMBED> tag anywhere on your Web page, you can use HTML to lay out the image window and controls any way you want.

3. Transfer the clip to the appropriate RealServer or Web server directory. When streaming from RealServer, you are now ready to test the clip, because the Web page already contains the link to the RealVideo clip.

➤ To create the Ram file when playing the clip back from a Web server:

1. With any text editor, open a new file and enter the URL your clip will have on the Web server, as in the following example:

   ```
   http://www.example.com/media/myclip.rm
   ```

2. Save this file as plain text with the file extension .rpm. This is your Ram file.
3. Transfer the Ram file to the appropriate directory on the Web server.
4. In your Web browser, click the link to verify that it works.

URL Reference

As explained in "RTSP and HTTP" on page 165, URLs to files and clips vary depending on what kind of file or clip you link to and what type of file contains the link. The following table provides a quick reference that shows which protocol is used for each of the various types of links. It also shows whether or not each link type should use RealServer's Ramgen utility.

Quick URL Reference

Link from	Link to	Located on	Protocol	Ramgen?	Reference
Web page	Media clip	RealServer	http://	**Yes**	page 166
		Web server	http://	No	page 170
Web page	SMIL file	RealServer	http://	**Yes**	page 166
		Web server	http://	No	page 170
Web page	Ram file (.ram or .rpm)	RealServer	http://	No	page 173
		Web server	http://	No	page 173
SMIL file	Media clip	RealServer	rtsp://	No	page 86
		Web server	http://	No	page 88
		Local host	file://	No	page 88
SMIL file	SMIL file	RealServer	rtsp://	No	page 86
		Web server	http://	No	page 88
		Local host	file://	No	page 88
Ram file (.ram or .rpm)	Media clip	RealServer	rtsp://	No	page 173
		Web server	http://	No	page 173
		Local host	file://	No	page 173
Ram file (.ram or .rpm)	SMIL file	RealServer	rtsp://	No	page 173
		Web server	http://	No	page 173
		Local host	file://	No	page 173

ADVANCED PRODUCTION TECHNIQUES

This appendix will help you get the most out of a broad range of features available in RealSystem and SMIL. Before putting into practice the production tips given here, you should have a solid understanding of SMIL, as described in Chapter 6.

Viewing SMIL Source Markup

RealPlayer 7 and later have a **View>Clip Source** command that shows the SMIL markup of the current presentation. The RealServer or Web server hosting the presentation sends the markup as an HTML page that opens in your default Web browser. This feature lets you examine SMIL presentations to learn how they are created.

Access to SMIL source information is denied for secure presentations that require a user name and password. The RealServer administrator may also disallow access to the SMIL source file, or allow access to the source file but conceal the full paths of clips. When access is allowed, the Web page showing the SMIL syntax includes a hypertext link for each clip in the presentation. Clicking a link takes you to a new Web page with information about the corresponding clip, including its size, buffer time, and streaming bit rate.

Smoothing Transitions Between Clips

By grouping clips played in sequence within a SMIL `<par>` tag, you can create smooth transitions between the clips. Normally, when clips play in sequence, each clip buffers data (its *preroll*) when it starts to play. By grouping the sequence within a `<par>` tag, though, RealSystem takes advantage of unused bandwidth to stream clips' preroll before the clips start to play. You can use this feature to mask preroll for high-bandwidth clips, for example, by streaming the preroll while low-bandwidth introductory clips play.

The following SMIL example, which omits the header that defines the region layout and base URL, shows how to mask preroll for high-bandwidth clips and create smooth transitions between the initial group of clips and the subsequent group of clips:

```
<body>
 <par>
  <seq>
   <par>
    <!-- group 1: introductory group masking preroll -->
    <audio src="intro.rm"/>
    <textstream src="titles.rt" region="left"/>
    <textstream src="credits.rt" region="right"/>
   </par>
   <par>
    <!-- group 2: main group with masked preroll -->
    <ref src="graphics.rp" region="left"/>
    <video src="story.rm" region="right"/>
   </par>
  </seq>
 </par>
</body>
```

Group 1 consists of two RealText clips and a RealAudio clip played in parallel. Because of the <seq> tag, the group 1 clips precede the RealPix and RealVideo clips in group 2. If the <seq> tag were the highest level of organization, RealServer would stream the group 1 clips without regard to group 2, streaming data for group 2 clips only after group 1 had finished playing. Viewers would thus experience a delay after the group 1 clips finished playing, while RealServer streams preroll for the group 2 clips.

You can eliminate this delay, however, by inserting an outer <par> tag, which is shown just below the <body> tag in the preceding example. This tag makes RealSystem treat groups 1 and 2 as one large parallel group with subgroups played in sequence. Although this does not affect the order in which the clips play, the parallel grouping makes RealSystem balance bandwidth among all of the clips. After it starts to stream the group 1 clips, RealServer makes use of unused bandwidth by streaming data for group 2 clips while the group 1 clips play. This masks the preroll for the group 2 clips.

Note

When you enclose clips in a <par> group, the individual clips do not appear in the RealPlayer playlist.

Additional Information

See "Buffering" on page 23 and "Timelines for Multiclip Presentations" on page 30.

Hiding Regions with z-index

RealPlayer creates all regions defined in a SMIL file's header section when it first reads the file. Thus, a single SMIL presentation cannot play clips in a certain set of regions and then destroy those regions and create different regions with a new layout. You can use the `<region>` tag's `z-index` parameter, however, to create transparent, overlaying regions that give the impression of regions appearing and disappearing.

The following SMIL header example creates a `left` region next to a `right` region. Both regions are displayed in portrait orientation and are more than twice as high as they are wide. A second set of regions, `top` and `bottom`, are stacked. These two regions have higher z-index values, meaning that they display in front of the `left` and `right` regions, as indicated in this example:

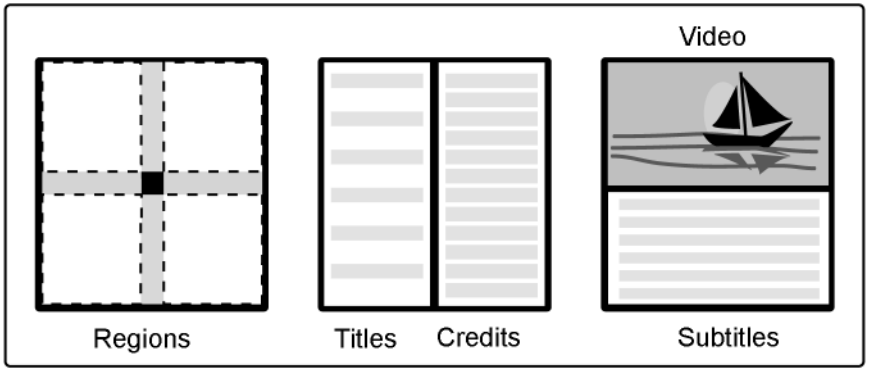
```
<head>
  <layout>
    <root-layout width="360" height="360"/>
    <!-- first two side-by-side regions -->
    <region id="left" top="10" left="10" width="165" height="340" z-index="0"/>
    <region id="right" top="10" left="185" width="165" height="340" z-index="1"/>
    <!-- second two stacked regions -->
    <region id="top" top="10" left="70" width="220" height="165" z-index="2"/>
    <region id="bottom" top="185" left="70" width="220" height="165" z-index="3"/>
  </layout>
</head>
```

As defined in the following SMIL body example, RealPix and RealText clips first play in the `left` and `right` regions, which appear behind the `top` and `bottom` regions. Because the overlaying `top` and `bottom` regions do not use background colors, they remain transparent until clips play in them. The introductory RealPix and RealText clips disappear when they finish playing, restoring to view the `root-layout` default background color, black. The group 2 clips, a RealVideo clip and a second RealText clip, then play in the `top` and `bottom` regions, as shown in this example:

```
<body>
  <par>
    <seq>
      <par>
        <!-- group 1: side-by-side titles and credits -->
        <ref src="titles.rp" region="left" fill="remove"/>
        <textstream src="credits.rt" region="right" fill="remove"/>
      </par>
      <par>
        <!-- group 2: stacked video and subtitles -->
        <video src="story.rm" region="top"/>
        <textstream src="subtitles.rt" region="bottom"/>
      </par>
    </seq>
  </par>
</body>
```

Although the `left`, `right`, `top`, and `bottom` regions exist from the moment the SMIL file starts to play, the use of `z-index`, `fill="remove"`, and default region transparency makes it appear as if the regions are created dynamically with each new set of clips. The following illustration shows the initial region creation, the first set of clips, and then the second set of clips playing in the RealPlayer window.

The Creation and Ordering of Transparent Regions



Additional Information

For more on `z-index`, see "Ordering Overlapping Regions with z-index" on page 106. "Laying Out Multiple Clips" on page 99 discusses SMIL layouts.

Writing Complex SMIL Switch Statements

The SMIL <switch> tag is a powerful feature with which you can specify options that each RealPlayer can choose between based on its preference settings and available bandwidth. "Switching Between Alternate Choices" on page 109 explains the basics of using the <switch> tag. This section provides tips on writing complex <switch> statements.

Switching with SureStream Clips

With RealAudio or RealVideo clips encoded for multiple bit rates with SureStream technology, you may or may not need to use the <switch> tag. The following guidelines will help you make this decision:

- When the presentation consists solely of a SureStream clip, simply link to that clip within the SMIL file. The clip then streams at the rate appropriate for RealPlayer's connection speed. You do not need to specify bandwidth choices with a <switch> tag.
- Use the <switch> tag when combining a SureStream clip with other clips encoded for single bandwidths. The SureStream clip is always used, but the <switch> group gives RealPlayer options for other clips. The following example illustrates a RealAudio SureStream clip and a choice between two RealPix presentations built for different bandwidths:

```
<par>
  <audio src="audio/newsong2.rm"/>
  <switch>
    <ref src="image/slideshow1.rp" system-bitrate="47000"/>
    <ref src="image/slideshow2.rp" system-bitrate="20000"/>
  </switch>
</par>
```

Additional Information

For more on SureStream, see "SureStream RealAudio and RealVideo" on page 27. Refer to "Step 3: Develop a Bandwidth Strategy" on page 22 for information on targeting certain network connection speeds.

Switching for Different Video Window Sizes

As described in "Different Window Sizes for Different Bandwidths" on page 52, you can create different-size versions of the same video, streaming a small

video window over slow modems and a larger window (or windows) over faster connections. Reducing the size for slower connections ensures that the video's frame rate and visual quality remain high. For example, you could create the three clips listed in the following table.

RealVideo Clips at Different Window Sizes

Clip Name	Window Size	SureStream Audiences	<switch> Attribute
videosmall.rm	176 x 132	28.8 and 56 Kbps Modems	20000
videomedium.rm	240 x 180	ISDN and corporate LANs	45000
videobig.rm	320 x 240	256, 384, and 512 Kbps DSL and cable modems	225000

Each <switch> tag test attribute uses the target bit rate of its clip's slowest SureStream stream. The <switch> tag then presents the three RealVideo choices to RealPlayer from fastest to slowest:

```
<switch>
  <video src="videobig.rm" system-bitrate="225000"/>
  <video src="videomedium.rm" system-bitrate="45000"/>
  <video src="videosmall.rm" system-bitrate="20000"/>
</switch>
```

Additional Information

Target bit rates are listed in the table "Maximum Streaming Rates" on page 24.

Switching with Multiple Test Attributes

You can use multiple <switch> test attributes to have RealPlayer choose clips based on both bandwidth and language. There are two ways to do this. In this first example, each audio clip choice has two test attributes—one for language and one for bandwidth. Both attributes must be viable for RealPlayer to choose the clip:

```
<switch>
  <!-- French language choices -->
  <audio src="sound/audio_fr2.rm" system-language="fr" system-bitrate="47000"/>
  <audio src="sound/audio_fr1.rm" system-language="fr" system-bitrate="20000"/>
  <!-- English language choices (default) -->
  <audio src="sound/audio_en2.rm" system-bitrate="47000"/>
  <audio src="sound/audio_en1.rm" system-bitrate="20000"/>
</switch>
```

Because RealPlayer evaluates the <switch> choices from top to bottom, selecting the first viable option, the last two choices do not have system-language options. This lets all RealPlayers other than those with French selected as their language preference choose between the two English-language clips.

The next example adds RealText clips in both French and English to the presentation possibilities. Here, <switch> statements are nested so that RealPlayers with French set as their language preference play the French RealText clip and choose from the set of French-language RealAudio clips, based on available bandwidth. All other RealPlayers play the English RealText clip and choose from the set of English-language RealAudio clips:

```
<switch>
  <!-- Choose French as the language -->
  <par system-language="fr">
    <textstream src="text/credits_fr.rt"/>
    <switch>
      <!-- Choose fast or slow bit rate for French audio -->
      <audio src="sound/audio_fr2.rm" system-bitrate="47000"/>
      <audio src="sound/audio_fr1.rm" system-bitrate="20000"/>
    </switch>
  </par>
  <!-- Choose English (default) as  the language -->
  <par>
    <textstream src="text/credits_en.rt"/>
    <switch>
      <!-- Choose fast or slow bit rate for English audio -->
      <audio src="sound/audio_en2.rm" system-bitrate="47000"/>
      <audio src="sound/audio_en1.rm" system-bitrate="20000"/>
    </switch>
  </par>
</switch>
```

Switching Presentations for Different RealPlayer Versions

Different versions of RealPlayer may have different features that affect which presentations they can play. RealPlayer G2 cannot use the caching and multiple-window features of RealPlayer 7 or later, for example. However, you can create different clips for different RealPlayer versions and then use a SMIL <switch> tag to enable each RealPlayer to choose the correct clips to play.

A <switch> tag can test for any number of RealPlayer versions from newest to oldest. It uses a system-required test attribute to specify which versions of

RealPlayer can play each choice. The last choice, which denotes the oldest version of RealPlayer you are considering, does not have a system-required attribute. A SMIL file with this type of <switch> tag takes the following form:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
 <body>
  <switch>
   <seq system-required="cv" cv:systemComponent="...attribute to test...">
    ...clips to play if RealPlayer satisfies the first test attribute...
   </seq>
   <seq system-required="cv" cv:systemComponent="...attribute to test...">
    ...clips to play if RealPlayer satisfies the second test attribute...
   </seq>
   <seq>
    ...clips to play for the default choice, which has no test attribute...
   </seq>
  </switch>
 </body>
</smil>
```

The <switch> tag in this example enables RealPlayer to choose among three <seq> groups, although RealPlayer could also choose among <par> groups or individual clips. The final, default choice must always be included and must not use the system-required test attribute.

Tip

You will need a separate computer for each version of RealPlayer you are testing. Multiple versions of RealPlayer cannot reside on the same computer.

➤ To write a <switch> tag to test for different versions of RealPlayer:

1. Declare the XML namespace in the <smil> tag.

   Although RealPlayer's implementation of SMIL does not typically need an XML namespace declaration, the use of systemComponent as a test attribute requires the following declaration:

   ```
   <smil xmlns:cv="http://features.real.com/systemComponent">
   ```

You must declare the namespace exactly as shown above. This declaration informs RealPlayer that the SMIL file uses the XML extension `systemComponent`. The declaration includes the following components:

`xmlns:`	Indicates an XML namespace declaration.
`cv=`	Namespace name defined by RealNetworks. The "cv" stands for "component version."
"*http://...*"	The quoted URL "`http://features.real.com/systemComponent`" is used by RealPlayer solely as an identifier that uniquely defines the namespace. RealPlayer does not contact this URL. The URL is valid, however, and displays an HTML page that describes the feature.

2. Write a `<switch>` tag that uses the proper `system-required` and `systemComponent` syntax. Within the `<switch>` tag, you include a `system-required` test attribute that takes the following form:

```
system-required="cv"
cv:systemComponent="http://features.real.com/?feature;player=6.0.7.x"
```

The components of the test attribute are as follows:

`system-required="cv"`	Refers to the "component version" namespace defined in the `<smil>` tag.
`cv:systemComponent=`	Defines a new feature test attribute. The case must match exactly.
"*http://...*"	The quoted URL "`http://features.real.com?feature;`" is the unique identifier for the component syntax. It matches the URL in the `<smil>` tag namespace declaration. RealPlayer does not request this URL.
`player=`	Specifies that `systemComponent` checks for a RealPlayer feature.
`6.0.7.x`	Indicates the version of RealPlayer to check for, such as `6.0.7.380` (the Gold version of RealPlayer 7). All RealPlayers with the given version number or higher choose this test attribute. You can find a RealPlayer's version number by choosing the **About RealPlayer** command on either the **Help** or **Apple** menu.

Note

The RealPlayer version number is the only part of the `systemComponent` syntax you should change.

Example 1: Caching Files for RealPlayer 7 or Later But Not RealPlayer G2

As explained in "Caching Files on RealPlayer" on page 132, RealPlayer 7 or later can cache downloaded image files, but RealPlayer G2 cannot. The following example is a SMIL <switch> tag that causes RealPlayer 7 or later to download and cache two logo files. If RealPlayer G2 requests this SMIL file, it simply downloads the logo files. This example is a variation of the SMIL file described in "Authoring SMIL Files for Caching" on page 134.

```
<smil xmlns:cv="http://features.real.com/systemComponent">
 <body>
  ....
  <switch>
   <seq system-required="cv"
    cv:systemComponent="http://features.real.com/?feature;player=6.0.7.380" >
    <!-- For RealPlayer 7 or later, download and cache these two logos" -->
    <img src="chttp://www.company.com/images/logo1.gif?bitrate=20000"
     region="logo1" fill="freeze"/>
    <img src="chttp://www.company.com/images/logo2.gif?bitrate=20000"
     region="logo2" fill="freeze"/>
   </seq>
   <seq>
    <!-- For RealPlayer G2, download these two logos" -->
    <img src="http://www.company.com/images/logo1.gif?bitrate=20000"
     region="logo1" fill="freeze"/>
    <img src="http://www.company.com/images/logo2.gif?bitrate=20000"
     region="logo2" fill="freeze"/>
   </seq>
  </switch>
 </body>
</smil>
```

When evaluating this <switch> statement, RealPlayer 7 or later chooses the first <seq> group. (The version number shown above, 6.0.7.380, is the Gold release of RealPlayer 7.) RealPlayer G2, on the other hand, ignores the first <seq> group and chooses the second <seq> group, which has no test attribute.

Example 2: Opening Multiple Windows for RealPlayer 7 or Later But Not RealPlayer G2

"Popping Up New RealPlayer Windows" on page 127 explains how RealPlayer 7 or later can open presentations in new windows. RealPlayer G2 uses only one window, however. The following SMIL <switch> tag presents RealPlayer with a choice between two RealText files. The first RealText file contains hyperlinks that launch clips in new RealPlayer windows. The second RealText file, written for RealPlayer G2, plays clips in the main RealPlayer window:

```
<smil xmlns:cv="http://features.real.com/systemComponent">
  <body>
    ....
    <switch>
      <!-- RealText clip to play with RealPlayer 7 or later. -->
      <textstream system-required="cv" cv:systemComponent=
        "http://features.real.com/?feature;player=6.0.7.380"
        src="rtsp://realserver.company.com/realtext/navigateRP7.rt
        region="text" fill="freeze"/>
      <!-- RealText clip to play with RealPlayer G2. -->
      <textstream
        src="rtsp://realserver.company.com/realtext/navigateG2.rt
        region="text" fill="freeze"/>
    </switch>
    ....
  </body>
</smil>
```

Displaying Captions for the Hearing Impaired

The RealPlayer preferences dialog includes an option to enable captions for the hearing impaired. To compose these captions, write a RealText clip that coordinates text with the audio timeline and then create a SMIL file that plays the two in parallel. As shown in the following example, the SMIL attribute system-captions="on" causes the captions to appear only if the viewer's preferences enable captions:

```
<smil>
  <head>
    <layout>
      <root-layout background-color="blue" width="310" height="200"/>
      <region id="video" width="176" height="132" left="67" top="5"/>
      <region id="captions" width="300" height="50" left="4" top="145"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="news.rm" region="video"/>
      <textstream src="captions.rt" region="captions" system-captions="on"/>
    </par>
  </body>
</smil>
```

In this example, the RealText clip (captions.rt) appears only for viewers who have enabled captions in RealPlayer. For all other viewers, the caption region displays only the root-layout region's background color. RealPlayer's caption setting does not affect the video clip, which has no system-captions value. The following sections explain how to fill the caption region or make it disappear when captions are turned off.

Tip

The system-captions attribute also works for <seq> and <par> groups. You can use system-captions="off" to cause a clip or group to play only when system captions are turned off in RealPlayer.

Additional Information

"Laying Out Multiple Clips" on page 99 discusses SMIL layouts. *RealText Authoring Guide* is available at **http://service.real.com/help/library/encoders.html**. The RealPlayer online help can assist you with turning captions on and off in the preferences.

Filling a Caption Region when Captions Are Turned Off

To avoid displaying a blank region when captions are turned off, you can display another file, such as an image file, in the caption region. The following excerpt is from a SMIL file that uses the same layout as the preceding example. Here, the parallel group includes an image clip with a system-captions="off" attribute. This causes the image to display only when captions are turned off in RealPlayer:

```
<par>
  <video src="news.rm" region="video"/>
  <textstream src="captions.rt" region="captions" system-captions="on"/>
  <img src="filler.gif" region="captions" system-captions="off"/>
</par>
```

Any clip—not just a static image—can fill the captions region when captions are turned off. However, the filler clip should be the same size as the RealText captions clip when the two display in the same region.

Resizing the Playback Area for Captions

As an extension to SMIL, RealPlayer supports the use of the system-captions attribute in layout tags. This lets you change layouts when captions are displayed. The following SMIL file centers a video region above a caption region only when captions are turned on for RealPlayer:

```
<smil>
  <head>
    <meta name="title" content="Closed-Captioned News"/>
    <!-- Layout used when captions are on. -->
    <layout system-captions="on">
      <root-layout background-color="blue" width="310" height="200"/>
      <region id="video" width="176" height="132" left="67" top="5"/>
      <region id="captions" width="300" height="50" left="4" top="145"/>
    </layout>
  </head>
  <body>
    <par>
      <video src="news.rm" system-captions="off"/>
      <video src="news.rm" region="video" system-captions="on"/>
      <textstream src="captions.rt" region="captions" system-captions="on"/>
    </par>
  </body>
</smil>
```

With this SMIL file, RealPlayer creates the layout and plays the RealText clip in parallel with the video only when captions are on. When captions are off, RealPlayer defines no regions and plays only the news.rm clip that includes the system-captions="off" attribute. When captions are off, the SMIL file is equivalent to the following:

```
<smil>
  <head>
    <meta name="title" content="Closed-Captioned News"/>
  </head>
  <body>
    <video src="news.rm" system-captions="off"/>
  </body>
</smil>
```

Note

Although the preceding example uses system-captions in the <layout> tag, you could use the attribute in <root-layout/> and <region/> tags as well to display or hide

individual regions based on RealPlayer's captions setting.

SMIL QUICK REFERENCE

Use this appendix as a reference tool when writing SMIL files. For complete information on SMIL and RealSystem, see Chapter 6. In the tables throughout this appendix, an asterisk (*) denotes a required attribute. Ellipses (...) indicate where nonessential information has been left out of examples.

Header Tags

<meta.../>

The header region's <meta.../> tags provide presentation information. A <meta.../> tag can also set a base URL for source clips in the SMIL file.

<meta.../> Tag Attributes

Attribute	Value	Function	Reference
content *	any value	Provides content for the name attribute.	page 118
name *	abstract	Gives presentation abstract.	page 118
	author	Lists name of presentation author.	page 118
	base	Sets base URL for source clips.	page 87
	copyright	Supplies presentation copyright.	page 118
	title	Gives presentation title, which displays in the RealPlayer run list.	page 118

Note

You can define other information attributes besides abstract, author, copyright, and title. See "Adding Presentation Information" on page 118.

Examples

```
<meta name="author" content="Jane Morales"/>
<meta name="title" content="Multimedia My Way"/>
<meta name="copyright" content="(c)1998 Jane Morales"/>
<meta name="base" content="rtsp://realserver.example.com/"/>
```

<layout>...</layout>

The <layout> and </layout> tags define the layout of visual clips such as video and animation. Within the tags, you define a root-layout region and separate regions for the clips. The layout must be defined in the SMIL header.

<root-layout.../>

Within <layout>...</layout>, <root-layout> sets the overall size of the playback area. Clips do not play in the root-layout area.

<root-layout.../> Tag Attributes

Attribute	Value	Function	Reference
background-color	color name or hex value	Sets root-layout color.	page 103
height *	pixels	Sets playback area height.	page 100
system-captions	on|off	Determines if root-layout is created based on system captions preference.	page 217
width *	pixels	Sets playback area width.	page 100

Example

```
<layout>
  <root-layout background-color="maroon" width="250" height="230"/>
  <region .../>
  <region .../>
</layout>
```

<region.../>

Within <layout>...</layout>, <region> tags define the size and placement (relative to root-layout) of each region used to play clips.

<region.../> Tag Attributes

Attribute	Value	Function	Reference
background-color	color name or hex value	Sets region background color.	page 103
fit	fill, hidden, meet, scroll, or slice	Determines how clip fits region when the two are different sizes.	page 104
height *	pixels or percentage	Sets region height.	page 101
id *	name	Provides target name for setting clips to play in the region.	page 101
left *	pixels or percentage	Sets region offset from the left side of the root-layout region.	page 101
system-captions	on|off	Determines if region is created based on system captions preference.	page 217
top *	pixels or percentage	Sets region offset from the top of the root-layout region.	page 101
width *	pixels or percentage	Sets region width.	page 101
z-index	number	Defines order of overlapping regions.	page 106

Example

```
<layout>
  <root-layout .../>
  <region id="video" top="5" left="5" width="240" height="180"
    background-color="blue" fit="fill" z-index="3"/>
  <region.../>
</layout>
```

Clip Source Tags

To define locations and timing attributes for clips within a presentation, use one of the following source tags:

```
<animation.../>
<audio.../>
<img.../>
<ref.../>
<text.../>
<textstream.../>
<video.../>
<RealAdInsert.../>
```

Except for <RealAdInsert.../>, the choice of tag does not affect playback. All clip source tags can use <ref.../>, for example. The <RealAdInsert.../> tag is not a SMIL tag, but a RealSystem tag that RealServer replaces with a SMIL tag for an advertising clip. It does not include a src attribute, but can include any other SMIL clip attribute. These attributes get passed to the replacement tag. See Chapter 9 for more on <RealAdInsert.../>.

Clip Source Tag Attributes

Attribute	Value	Function	Reference
abstract	abstract	Gives abstract for clip.	page 119
author	name	Lists author for clip.	page 119
begin	h, min, s, or ms	Delays normal playback time.	page 92
clip-begin	h, min, s, or ms	Specifies clip's internal timing mark where playback begins.	page 93
clip-end	h, min, s, or ms	Specifies clip's internal timing mark where playback ends.	page 93
copyright	copyright	Lists copyright information for clip.	page 119
dur	h, min, s, or ms	Sets total time clip plays. Do not use with end.	page 94
end	h, min, s, or ms	Sets end time for clip relative to begin time. Do not use with dur.	page 92
fill	freeze or remove	Freezes or removes (default) clip from screen when it stops playing.	page 95
id	name	Names clip for reference by other SMIL elements.	page 90
region	<region.../> tag id name	Assigns clip to a region.	page 107
repeat	integer or indefinite	Repeats the clip the specified number of times, or indefinitely.	page 97
src *	URL	Provides full or relative URL for clip.	page 85

(Table Page 1 of 2)

Clip Source Tag Attributes (continued)

Attribute	Value	Function	Reference
system-bitrate	bits per second	Sets bit rate attribute for <switch>...</switch>.	page 110
system-captions	on|off	Determines if clip plays based on system captions preference.	page 215
system-language	language code	Sets language attribute for <switch>...</switch>.	page 109
title	title	Lists title for clip.	page 119

(Table Page 2 of 2)

Examples

```
<video src="rtsp://realserver.example.com/media/video2.rm" region="video"
  begin="40s" clip-begin="5100ms" clip-end="4.5min" fill="freeze"/>

<audio src="rtsp://realserver.example.com/media/music.rm"
  dur="10.5s" repeat="5"/>
```

Image Source Tag Options

For a still image, you can include the following options in the image source tag. Note that these options are extensions to SMIL that work only with RealPlayer.

Image Source Tag Options

Option	Value	Function	Reference
bgcolor	color name or hex value	Changes GIF transparency to color.	page 125
bitrate	bits per second (bps)	Sets streaming bit rate.	page 124
reliable	true	Guarantees reliable image transmission.	page 126
target	_browser	Makes Web browser the URL target.	page 125
	_player	Sets RealPlayer as command target.	page 126
url	URL	Sends URL to browser on mouse click.	page 125
	command:play()	Gives play command on mouse click.	page 126
	command:pause()	Gives pause command on mouse click.	
	command:seek(*time*)	Makes RealPlayer seek to specified time in timeline on mouse click. Time has dd:hh:mm:ss.xyz format.	
	command:stop()	Gives stop command on mouse click.	

A question mark operator (?) separates image options from the image URL in the clip source tag. Additional options preceded by ampersands (&) can follow in this format:

```
<img src= "URL?option=value&option=value"/>
```

Note

Image options and values are not in quotation marks because they are part of the quoted src value.

Examples

```
<img src="ad1.gif?bitrate=1000&bgcolor=blue"/>

<img src="ad.gif?url=http://www.example.com&target=_browser" .../>

<img src="stop.gif?url=command:stop()&target=_player" .../>

<img src="seek.gif?url=command:seek(1:35.4)&target=_player" .../>
```

Group Tags

<par>...</par>

The <par> and </par> tags make enclosed clips play in parallel. No attributes are required for a <par> tag.

<par> Tag Attributes

Attribute	Value	Function	Reference
abstract	abstract	Gives abstract for group.	page 119
author	name	Lists author for group.	page 119
begin	h, min, s, or ms	Delays normal group playback time.	page 93
copyright	copyright	Lists copyright for group.	page 119
dur	h, min, s, or ms	Sets total time group plays.	page 94
end	h, min, s, or ms	Sets end time for group relative to begin time. Do not use with endsync.	page 93
endsync	first	Ends group when first clip ends.	page 90
	last	Ends group when last clip ends.	
	id(*clip id*)	Ends group when specified clip ends.	
id	name	Names group for reference by other SMIL elements.	page 116

(Table Page 1 of 2)

<par> Tag Attributes (continued)

Attribute	Value	Function	Reference
repeat	integer or indefinite	Repeats the group the specified number of times, or indefinitely.	page 97
system-bitrate	bits per second	Sets bit rate attribute for <switch>...</switch>.	page 110
system-captions	on|off	Determines if group plays based on system captions preference.	page 215
system-language	language code	Sets language attribute for <switch>...</switch>.	page 109
title	title	Lists title for group.	page 119

(Table Page 2 of 2)

Examples

```
<par>
  <video src="videos/newsong.rm"/>
  <textstream src="lyrics/newsong.rt"/>
</par>
```

```
<par endsync="id(text)" repeat="2" begin="4s">
  <video src="videos/newsong.rm"/>
  <textstream id="text" src="lyrics/newsong.rt"/>
</par>
```

<seq>...</seq>

The <seq> and </seq> tags group clips that play in sequence. No attributes are required for a <seq> tag.

<seq> Tag Attributes

Attribute	Value	Function	Reference
begin	h, min, s, or ms	Delays normal group playback time.	page 93
dur	h, min, s, or ms	Sets total time group plays.	page 94
end	h, min, s, or ms	Sets end time for group relative to begin time.	page 93
id	name	Names group for reference by other SMIL elements.	page 116
repeat	integer or indefinite	Repeats the group the specified number of times, or indefinitely.	page 97

(Table Page 1 of 2)

<seq> Tag Attributes (continued)

Attribute	Value	Function	Reference
system-bitrate	bits per second	Sets bit rate attribute for <switch>...</switch>.	page 110
system-captions	on|off	Determines if group plays based on system captions preference.	page 215
system-language	language code	Sets language attribute for <switch>...</switch>.	page 109

(Table Page 2 of 2)

Example

```
<seq repeat="3">
  <audio src="rtsp://realserver.example.com/one.rm"/>
  <audio src="rtsp://realserver.example.com/two.rm"/>
</seq>
```

<switch>...</switch>

The <switch> and </switch> tags contain clips or groups of clips that RealPlayer chooses between based on its available bandwidth or language preference. Clips or groups that RealPlayer evaluates must include a system-bitrate or system-language attribute.

<switch> Tag Attributes

Attribute	Value	Function	Reference
id	name	Names group as a link target for other SMIL files.	page 116

Additional Information

See "Switching Between Alternate Choices" on page 109.

Example

```
<switch>
  <audio src="french/seattle.rm" system-language="fr"/>
  <audio src="german/seattle.rm" system-language="de"/>
  <audio src="english/seattle.rm"/>
</switch>
```

Hyperlink Tags

<a>...

The <a>... tags work like HTML hyperlink tags to connect a media source clip to another clip. But whereas you enclose text between <a> and in HTML, you enclose a media source tag between <a> and in SMIL.

<a.../> Tag Attributes

Attribute	Value	Function	Reference
href *	URL	Sets hyperlink URL.	page 112
show	new	Opens URL in browser while RealPlayer continues.	page 112
	pause	Opens URL in browser and pauses RealPlayer.	
	replace (default)	Opens URL in RealPlayer.	

Example

```
<a href="http://www.example.com/index.htm" show="new">
  <video src="video.rm" region="videoregion"/>
</a>
```

<anchor.../>

An <anchor> tag can define a hot spot hyperlink that can be temporal as well as spatial. It fits within a media source tag:

```
<video ...>
  <anchor .../>
</video>
```

Note that here the <video> source tag does not end with a forward slash as it normally does. Instead, a </video> tag follows it and the <anchor.../> tag.

<anchor.../> Tag Attributes

Attribute	Value	Function	Reference
begin	h, min, s, or ms	Defines when hot spot becomes active or when linked media clip begins in its timeline.	page 116 page 118
coords	pixels or percentages	Defines boundary for hot spot as left, top, right, and bottom offset from source clip.	page 114
end	h, min, s, or ms	Determines when hot spot deactivates.	page 116

(Table Page 1 of 2)

<anchor.../> Tag Attributes (continued)

Attribute	Value	Function	Reference
href *	URL	Sets hyperlink URL.	page 114
id	name	Defines target for use by other hypertext links.	page 118
show	new	Opens URL in browser while RealPlayer continues playback.	page 112
	pause	Opens URL in browser and pauses RealPlayer.	
	replace (default)	Opens URL in RealPlayer.	

(Table Page 2 of 2)

Examples

```
<video src="video.rm" region="videoregion">
  <anchor href="rtsp://realserver.example.com/vid2.rm" coords="20,40,80,120"/>
</video>
```

```
<video src="video.rm" region="videoregion">
  <anchor href="http://www.example.com" show="pause" coords="10,30,70,100"
    begin="5s" end="45s"/>
</video>
```

Pop-Up Window Commands

In a SMIL hyperlink, you can include a `command:openwindow` argument with the following options to open the link target in a new RealPlayer window. These extensions to SMIL work only with RealPlayer 7 and later.

Pop-up Window Attributes

Attribute	Value	Function	Reference
(name) *	_new	Opens URL in a new RealPlayer window.	page 128
	_blank		
	_self	Opens URL in the current window.	
	_current		
	name	Opens URL in a new RealPlayer window that has the given name.	
(URL) *	URL	Provides URL for the new clip or presentation.	page 128

(Table Page 1 of 2)

Pop-up Window Attributes (continued)

Attribute	Value	Function	Reference
autosize	true	Puts new window in autosize mode.	page 128
	false	Opens new window in compact mode. (Default)	
zoomlevel	normal	Plays clip at its normal size. (Default)	
	double	Doubles clip size.	
	full	Plays clip at full-screen.	
ontopwhile playing	true	Keeps new window on top of other windows.	
	false	Lets user control window placement. (Default)	

(Table Page 2 of 2)

The syntax for a pop-up window in an <a> or <anchor> tag looks like the following, with *name* and *URL* coming first and second as single statements rather than as attribute=value pairs:

```
<a href="command:openwindow(name, URL, playmode=value, ...)">...</a>
```

Examples

```
<a href="command:openwindow(_new, rtsp://realserver.example.com/animals.rm,
    autosize=true, ontopwhileplaying=true)">
  <img src="launch_button.gif" region="button"/>
</a>
```

```
<video src="rtsp://realserver.example.com/video/video1.rm" region="video">
  <anchor href="command:openwindow(popup,
    rtsp://realserver.example.com/video/video2.rm, zoomlevel=double)"
    coords="0,0,25%,25%"/>
</video>
```

SMIL LANGUAGE CODES

As "Setting Language Choices" on page 109 explains, SMIL can list different language choices that RealPlayer chooses from based on its language preference. The following table lists the codes you can use in a SMIL file to indicate clips created for specific languages.

Code	Language
af	Afrikaans
sq	Albanian
ar-iq	Arabic (Iraq)
ar-dz	Arabic (Algeria)
ar-bh	Arabic (Bahrain)
ar-eg	Arabic (Egypt)
ar-jo	Arabic (Jordan)
ar-kw	Arabic (Kuwait)
ar-lb	Arabic (Lebanon)
ar-ly	Arabic (Libya)
ar-ma	Arabic (Morocco)
ar-om	Arabic (Oman)
ar-qa	Arabic (Qatar)
ar-sa	Arabic (Saudi Arabia)
ar-sy	Arabic (Syria)
ar-tn	Arabic (Tunisia)
ar-ae	Arabic (U.A.E.)
ar-ye	Arabic (Yemen)
eu	Basque
bg	Bulgarian
ca	Catalan

Code	Language
zh-hk	Chinese (Hong Kong)
zh-cn	Chinese (People's Republic)
zh-sg	Chinese (Singapore)
zh-tw	Chinese (Taiwan)
hr	Croatian
cs	Czech
da	Danish
nl	Dutch (Standard)
nl-be	Dutch (Belgian)
en	English
en-au	English (Australian)
en-bz	English (Belize)
en-gb	English (British)
en-ca	English (Canadian)
en	English (Caribbean)
en-ie	English (Ireland)
en-jm	English (Jamaica)
en-nz	English (New Zealand)
en-za	English (South Africa)
en-tt	English (Trinidad)
en-us	English (United States)

Code	Language
et	Estonian
fo	Faeroese
fi	Finnish
fr-be	French (Belgian)
fr-ca	French (Canadian)
fr-lu	French (Luxembourg)
fr	French (Standard)
fr-ch	French (Swiss)
de-at	German (Austrian)
de-li	German (Liechtenstein)
de-lu	German (Luxembourg)
de	German (Standard)
de-ch	German (Swiss)
el	Greek
he	Hebrew
hu	Hungarian
is	Icelandic
in	Indonesian
it	Italian (Standard)
it-ch	Italian (Swiss)
ja	Japanese
ko	Korean
ko	Korean (Johab)
lv	Latvian
lt	Lithuanian
no	Norwegian
pl	Polish
pt-br	Portuguese (Brazilian)
pt	Portuguese (Standard)
ro	Romanian
sr	Serbian
sk	Slovak

Code	Language
sl	Slovenian
es-ar	Spanish (Argentina)
es-bo	Spanish (Bolivia)
es-cl	Spanish (Chile)
es-co	Spanish (Colombia)
es-cr	Spanish (Costa Rica)
es-do	Spanish (Dominican Republic)
es-ec	Spanish (Ecuador)
es-sv	Spanish (El Salvador)
es-gt	Spanish (Guatemala)
es-hn	Spanish (Honduras)
es-mx	Spanish (Mexican)
es-ni	Spanish (Nicaragua)
es-pa	Spanish (Panama)
es-py	Spanish (Paraguay)
es-pe	Spanish (Peru)
es-pr	Spanish (Puerto Rico)
es	Spanish (Spain)
es-uy	Spanish (Uruguay)
es-ve	Spanish (Venezuela)
sv	Swedish
sv-fi	Swedish (Finland)
th	Thai
tr	Turkish
uk	Ukrainian
vi	Vietnamese

FILE TYPE REFERENCE

The following tables provide a quick reference to file types commonly used in RealSystem streaming. This is not a definitive list of all file types, though. Plug-in technology allows RealSystem to stream virtually any file type.

RealSystem Standard Streaming Clip Types

Extension	File Type	Reference
.rm or .ra	RealAudio	"Understanding RealAudio" on page 33
.rm	RealVideo	"Understanding RealVideo" on page 47
.rp	RealPix streaming image markup	"Images" on page 19
.rpa	RealPix ad rotation	"Displaying Banner Ads" on page 158
.rt	RealText streaming text	"Text" on page 21
.swf	Flash Player file	"Producing Animation" on page 65

RealSystem Information Files

Extension	File Type	Reference
.ram	Ram file to launch RealPlayer	"Creating a Ram File Manually" on page 173
.rpm	Ram file for embedded presentations	
.smil, .smi	SMIL file for layout and timing	"General SMIL Rules" on page 83

Image Files Types Playable Directly in RealPlayer and RealPix

Extension	File Type	Reference
.gif	GIF87, GIF89, or animated GIF image	"Images" on page 19
.jpg	JPEG (nonprogressive) image	
.png	PNG image	

GLOSSARY

A

Advertising Application
A RealServer option that inserts ads into SMIL-based presentations. SMIL files use `<RealAdInsert/>` tags to indicate ad placement.

artifact
A visual imperfection in an encoded video clip. Too many artifacts can make the video look blocky.

B

bandwidth
The upper limit on the amount of data, typically expressed as kilobits per second (Kbps), that can pass through a network connection.

banner ad
An ad that appears alongside a requested clip or presentation. A banner ad can also rotate, making new ad images appear at regular intervals.

bit
The smallest unit of measure of data in a computer. A bit has a binary value, either `0` or `1`.

bit rate
A measure of bandwidth, expressed as the number of bits transmitted per second. A 28.8 Kbps modem, for example, can transmit or receive around 29,000 bits per second.

broadcast
To deliver a presentation, whether live or prerecorded, in which all viewers join the presentation in progress. Contrast to *on-demand*.

buffering
The receiving and storing of data before it is played back. A clip's initial buffering is called *preroll*. After this preroll, excessive buffering may stall the presentation.

byte
A common measurement of data. One byte consists of 8 bits.

C

cable modems
Devices that allow rapid transmission and reception of data over television cable. They are digital devices, unlike dial-up modems, which transmit analog data.

CBR
Constant Bit Rate. A type of RealVideo encoding in which all parts of the video play back at the same bit rate. Contrast to *VBR*.

CHTTP
A version of HTTP supported by RealPlayer. Files designated with `chttp://` are downloaded through HTTP and stored in RealPlayer's cache.

client
A software application that receives data from a server. A Web browser is a client of a Web server. RealPlayer is a client of RealServer.

clip
A media file within a presentation. Clips typically have an internal timeline, as with RealAudio and RealVideo.

codec
Coder/decoder. Codecs convert data between uncompressed and compressed formats, reducing the bandwidth a clip consumes.

D

download
To send a file over a network with a nonstreaming protocol such as HTTP. Contrast to *stream*.

DSL
Digital Subscriber Line. A technology for transmitting digital data over a regular telephone line at speeds much faster than dial-up modems.

duress stream
A low-bandwidth SureStream audio or video stream that RealServer uses if a connection's available bandwidth drops greatly.

E

encoding
Converting a file into a compressed, streaming format. For example, you can encode WAV files as RealAudio clips.

F

Flash
A software application and an animation format created by Macromedia. RealPlayer can play Flash animations and stream them in parallel with other clips, such as RealAudio clips.

Flash Player file
A compressed Flash file format (file extension `.swf`) suitable for streaming. To stream Flash, you export the Flash Player file and tune it so that it plays well in RealPlayer.

fps
Frames Per Second. The number of video frames that displays each second in a streaming video clip.

frequency response
A measure of audio clip quality. The higher a clip's frequency response, the more frequencies it can faithfully reproduce.

H

HTTP
Hypertext Transport Protocol. The protocol used by Web servers to communicate with Web browsers. In contrast, RealServer streams clips to RealPlayer with RTSP. See also *CHTTP*.

I

interstitial ad
An advertisement that interrupts playback of a requested clip. It is like a commercial break on television.

ISDN
Integrated Services Digital Network. Technology that makes digital data connections at 64 or 112 Kbps possible over telephone lines.

ISP
Internet Service Provider. A company that provides access to the Internet. Many ISPs have RealServer available to stream media clips.

K

kilobit (Kb)
A common unit of data measurement equal to 1024 bits. A kilobit is usually referred to in the context of bit rate per unit of time, such as kilobits per second (Kbps).

kilobyte (KB)
A common unit of data measurement equal to 1024 bytes or 8 kilobits.

L

LAN
Local Area Network. A computer network confined to a local area, such as a single building. LANs vary in speed, with bandwidth shared among all networked devices.

lossy
A compression scheme that lowers clip size by discarding nonessential data from the source file. Both RealAudio and RealVideo are lossy.

M

mouseover
The action of moving a computer screen pointer over an interactive area. An animated button may change appearance on a mouseover, for example.

0

on-demand
A type of streaming in which a clip plays from start to finish when a user clicks a link. Most clips are streamed this way. Contrast to *broadcast*.

P

PNA
A proprietary protocol RealServer uses for backward compatibility with RealSystem 3 through 5. URLs using PNA start with `pnm://`.

port
A connection to a server, designated by a number such as 8080. RealServer uses different ports for the RTSP, HTTP, and PNA protocols.

preroll
Buffering that occurs before a clip plays back. Preroll should be no more than 15 seconds.

presentation
A group of clips coordinated through SMIL and streamed from RealServer to RealPlayer.

R

RDT
The proprietary data package protocol RealServer uses (along with RTSP) when communicating with RealPlayer. Contrast to *RTP*.

RealAudio
A RealSystem clip type for streaming audio over a network. RealAudio clips use the `.rm` extension.

RealPix
A RealSystem clip type (file extension `.rp`) for streaming still images over a network. RealPix uses a markup language for creating special effects such as fades and zooms.

RealPlayer
RealNetworks client software designed to play multimedia presentations streamed by RealServer or a Web server.

RealProducer
The primary RealNetworks tool for encoding RealAudio and RealVideo clips.

RealServer
RealNetworks server software used to stream multimedia presentations to RealPlayer.

RealServer administrator
The person in charge of setting up and running RealServer.

RealSlideshow
A RealNetworks tool for creating streaming slideshows based on the RealPix markup.

RealSystem
The RealNetworks system for streaming media such as RealAudio and RealVideo clips over a network. It consists of RealServer, RealPlayer, and various production tools.

RealText
A RealSystem clip type (file extension `.rt`) for streaming text over a network. It uses a markup language for formatting text.

real-time
Delivered as it occurs. For example, a live event is streamed across a network in a real-time broadcast.

RealVideo
A RealSystem clip type for streaming video over a network. RealVideo clips use the extension `.rm`.

rotating banner ads
Banner advertisements that change at a specified interval, such as a new ad banner every 30 seconds.

RTP
Real-Time Protocol. The open, standards-based data package protocol RealServer uses (along with RTSP) to communicate with RTP-based clients. Contrast to *RDT*.

RTSP
Real-Time Streaming Protocol. An open, standards-based control protocol that RealServer uses to stream clips to RealPlayer or any RTP-based client. Contrast to *HTTP*.

S

server
1. A software application, such as a Web server or RealServer, that sends requested data over a network.
2. A computer that runs server software.

Shockwave Flash
See *Flash Player file*.

SMIL
Synchronized Multimedia Integration Language. A markup language for specifying how and when each clip plays within a presentation. SMIL files use the extension `.smil`.

stream
1. To send a media clip over a network so that it begins playing back as quickly as possible.
2. A flow of a single type of data, measured in kilobits per second (Kbps). A RealVideo clip's soundtrack is one stream, for example.

SureStream
A RealNetworks technology that enables a RealAudio or RealVideo clip to stream at multiple bit rates. Available only in RealSystem.

U

URL
Uniform Resource Locator. A location description that enables a Web browser or RealPlayer to receive a clip stored on a Web server or RealServer.

V

VBR
Variable Bit Rate. A type of RealVideo encoding that enables RealPlayer to play different parts of the video at different bit rates, even though the video is being streamed at a constant rate. Contrast to *CBR*.

INDEX